.NET FRAMEWORK SECURITY CHECKLIST

Version 1, Release 2.1

21 September 2007

Developed by DISA for the DOD

This page is intentionally left blank.

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1 OVERVIEW 1
1.2 ORGANIZATION OF THE CHECKLIST 1
1.3 SUPPORTED VERSIONS 2
1.4 DOCUMENT EFFECTIVE DATE 2
1.5 REVIEW METHOD 2
1.6 REFERENCED DOCUMENTS 2

2. .NET FRAMEWORK SRR RESULTS REPORT 3

2.1 SITE INFORMATION 3
2.2 SYSTEM INFORMATION 4

3. .NET FRAMEWORK SECURITY CHECKLIST AND PROCEDURES 13

3.1 REVIEWER NOTES 13
3.2 IAVM COMPLIANCE 13
3.3 DETERMINING WHICH VERSIONS OF THE .NET FRAMEWORK ARE INSTALLED 14
3.4 REVIEWER INTERFACES 14
3.4.1 Using the Microsoft .NET Framework Configuration Tool MSCORCFG.MSC 14
3.4.2 Using the .NET Framework Code Access Security (CAS) Policy Tool CASPOL.EXE 19
3.4.3 Using the Strong Name Tool SN.EXE 20
3.4.4 Using the Software Publishing State Tool SETREG.EXE 20
3.4.5 Review Results 20
3.4.6 Version-specific Vulnerabilities 20
3.4.7 Assemblies, Evidence, Permission Sets, and Code Groups 20
3.4.8 Determining Effective Permissions 43
3.4.9 .NET Security Framework Checks and Procedures 43

4. VMS 6.0 PROCESS AND PROCEDURES 88

4.1 SYSTEM ADMINISTRATOR 88
4.2 REVIEWER 88

LIST OF TABLES

Table 1-1. Resources 2
Table 2-1. System Information 4
Table 2-2. Summary of Database SRR Findings By Category 4

1. INTRODUCTION

1.1 Overview

The .NET Framework Security Readiness Review (SRR) targets conditions that undermine the integrity of security, contribute to inefficient security operations and administration, or may lead to interruption of production operations. Additionally, the review ensures the site has properly installed and implemented the .NET environment and that it is being managed in a way that is secure, efficient, and effective. The items reviewed are based the NSA guide, *Guide to Microsoft .NET Framework Security*. The results of the review should be recorded in the SRR Results section with the following status designations: F- Finding, N/F- Not A Finding, N/A- Not Applicable, MR -Manual Review, or NR – Not Reviewed.

DISA Field Security Operations has assigned a level of urgency to each finding based on Chief Information Officer (CIO) established criteria for certification and accreditation. All findings are based on regulations and guidelines. All findings require correction by the host organization. Category I findings are any vulnerabilities that provide an attacker immediate access into a machine, super user access, or access that bypasses a firewall. Category II findings are any vulnerabilities that provide information that has a high potential of giving access to an intruder. Category III findings are any vulnerabilities that provide information that potentially could lead to compromise. Category IV vulnerabilities, when resolved, will prevent the possibility of degraded security.

NOTE: Security patches required by the DOD IAVM process are reviewed during an operating system security review. Information for security patch compliance is available in Appendix A of this .Net Framework Security Checklist.

1.2 Organization of the Checklist

The .NET Framework Security Checklist is composed of five major sections and three appendices. The organizational breakdown proceeds as follows:

Section 1	Introduction
	This section contains summary information about the sections and appendices that comprise the *.NET Framework Security Checklist,* and defines its scope. Supporting documents consulted are listed in this section.
Section 2	.NET SRR Result Report
	This section is the matrix that allows the reviewer to document vulnerabilities discovered during the SRR process. This section is used for a .NET Framework Security review.
Section 3	.NET Checklist Procedures

This section documents the procedures that instruct the reviewer on how to perform an SRR using the manual procedures, and how to interpret the resulting information for vulnerabilities. Each procedure maps to a specific vulnerability listed in Section 2.

1.3 Supported Versions

The vulnerabilities discussed in Sections 2 and 3 of this document are applicable to .NET Framework 1.0, 1.1, and 2.0.

1.4 Document Effective Date

This document is effective as of April, 14 2006. This document will be updated as needed.

1.5 Review Method

To perform a successful Security Readiness Review (SRR), a manual process must be employed. There are currently no automated tools to check for compliance with this checklist.

Since each version of the .NET Framework is configured separately a SRR must be performed against each version of the .NET framework that is installed on the system. Refer to Section 3.3 of this document for instructions on determining which versions of the .NET Framework are installed on the system.

1.6 Referenced Documents

The following table enumerates the documents and resources consulted:

Date	Document Description
22 Sep 2004	*Guide to Microsoft .NET Framework Security,* NSA SNAC, v1.4

Table 1-1. Resources

2. .NET FRAMEWORK SRR RESULTS REPORT

Unclassified UNTIL FILLED IN
CIRCLE ONE
FOR OFFICIAL USE ONLY (mark each page)
CONFIDENTIAL and SECRET (mark each page and each finding)

Classification is based on classification of system reviewed:
Unclassified System = FOUO Checklist
Confidential System = CONFIDENTIAL Checklist
Secret System = SECRET Checklist
Top Secret System = SECRET Checklist

This checklist becomes effective September 30, 2005

Reviewer: _____ Date:

System: Type of Review (Remote, Sample, Full):

Finding Totals:	**Comments:**
Category I: _____	
Category II: _____	
Category III:	
Total:	

2.1 Site Information

Site: _____

System Administrator Information:
Name: _____
E-mail Address: _____
Phone # (Commercial): ( ) _____ DSN: _____

IAO Information:
Name: _____
E-Mail Address _____
Phone # (Commercial) _____ DSN: _____

2.2 System Information

System Detail	
System ID or Host Name	
Hardware Platform	
Operating System	
Operating System Version	
MAC Level	
Confidentiality Level	

Table 2-1. System Information

Summary of .Net Framework SRR Findings By Category		
Category	**Total Possible Findings**	**Actual Findings**
Category I	1	
Category II	36	
Category III	8	
Total Findings	45	0

Table 2-2. Summary of .Net Framework SRR Findings By Category

(A=Completely Automated, MR = Partially Automated (Manual Review), NC=Can Be Automated, NR = Not Reviewed (Cannot be Automated)).

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**CAT**
3.4.9.1	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *File IO* permission is granted with Path Discovery. The *File IO* permission is granted with *unrestricted="true"*. The *File IO* permission is granted to unrestricted paths.	APPNET0001	File IO Permission	CAT III
3.4.9.2	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Isolated Storage* permission is granted with *unrestricted="true"*. The *Isolated Storage* permission is granted with *allowed=Administrator Isolated Storage By User* and is not monitored. The *Isolated Storage* permission is granted with *allowed=Assembly Isolation Storage By Roaming User* and is not monitored.	APPNET0003	Isolated Storage Permission	CAT III
3.4.9.3	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *User Interface* permission is granted with *unrestricted="true"*. The *User Interface* permission is granted with *Window=AllWindowsandEvents*. An unauthorized *User Interface* permission with *Windows=SafeTopLevelWindows* is granted.	APPNET0004	User Interface Permission (Windowing)	CAT II
3.4.9.4	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *User Interface* permission is granted with *unrestricted="true"*. The *User Interface* permission is granted with access other than *Clipboard=NoClipboard*.	APPNET0005	User Interface Permission (Clipboard)	CAT II
3.4.9.5	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Reflection* permission is granted with *unrestricted="true"*. The *Reflection* permission is granted with *Flags="Member"*. The *Reflection* permission is granted with *Flags="Type" to* software that is not a confirmed engineering tool or software interoperability service.	APPNET0006	Reflection Permission	CAT III

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**CAT**
3.4.9.6	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Printing* permission is granted with *unrestricted="true"*. The *Printing* permission is granted with *Level=AllPrinting*.	APPNET0007	Printing Permission	CAT III
3.4.9.7	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *DNS* permission is granted with *unrestricted="true*" to assemblies that do not originate within the local network.	APPNET0008	DNS Permission	CAT II
3.4.9.8	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Socket* permission is granted with *unrestricted="true"*. The *Socket* permission is granted to code that does not provide networking service. The *Socket* permission is granted to code that originates from an external network.	APPNET0009	Socket Access Permission	CAT II
3.4.9.9	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Web* permission is granted with *unrestricted="true"*. The *Web* permission is granted to specific URLs that are not documented and approved for sharing data.	APPNET0010	Web Access Permission	CAT II
3.4.9.10	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Message Queue* permission is granted with *unrestricted="true"*. The *Message Queue* permission is granted with *access=Administer* to a queue that is not used by a confirmed administrative tool. The *Message Queue* permission is granted with *access=Browse* to all queues code that is not a verified administrative tool.	APPNET0011	Message Queue Permission	CAT II
3.4.9.11	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Service Controller* permission is granted with *unrestricted="true"*. The *Service Controller* permission is granted to a service whose code is not documented as having the same level of trust and value as the service(s) itself.	APPNET0012	Service Controller Permission	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**CAT**
3.4.9.12	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The database client permission is granted with *unrestricted="true"*. The database permission to specific providers is granted to unauthorized code.	APPNET0013	Database Permission	CAT III
3.4.9.13	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *unrestricted="true"*. The *Security* permission is granted with *Flags="Infrastructure"* (*Extend infrastructure*) to any code that has not been verified as having complete control over message processing.	APPNET0014	Security Permission (Extend Infrastructure)	CAT II
3.4.9.14	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="RemotingConfiguration"* to code that is not highly trusted (has a strong name with a public key associated with a local entity) and that does not require network access.	APPNET0015	Security Permission (Enable Remoting Configuration)	CAT II
3.4.9.15	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="SerializationFormatter"* to code that is not authorized as an extension to the CLR's trusted library base.	APPNET0016	Security Permission (Enable Serialization Formatter)	CAT II
3.4.9.16	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="ControlThread"* to code that is not Fully Trusted (a member of the FullTrust permission set).	APPNET0017	Security Permission (Enable Thread Control)	CAT II
3.4.9.17	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="ControlPrincipal"* (*Allow principal control*) to code that is not documented as being as trusted as the most privileged system user account.	APPNET0018	Security Permission (Allow Principal Control)	CAT II
3.4.9.18	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="Execution"* (*Allow assembly execution*) to unauthorized code.	APPNET0019	Security Permission (Enable Assembly Execution)	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**CAT**
3.4.9.19	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="Skip Verification"* (*Skip verification*) to code that is not highly trusted (has a strong name with a public key associated with a trusted party).	APPNET0020	Security Permission (Skip Verification)	CAT II
3.4.9.20	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="UnmanagedCode"* (*Allow calls to unmanaged assemblies*).	APPNET0021	Security Permission (Allow Calls to Unmanaged Assemblies)	CAT II
3.4.9.21	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="ControlPolicy"* (*Allow Policy Control*) to code that is not a highly trusted (has a strong name with a public key associated with a local entity) administrative tool.	APPNET0022	Security Permission (Allow Policy Control)	CAT II
3.4.9.22	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="ControlDomainPolicy"* (*Allow Domain Policy Control*) to code that is not highly trusted (has a strong name with a public key associated with a local entity) and is not a custom Runtime Host application that implements organizational policy using the AppDomain CAS policy level or to code that does not require the dynamic launching of applications that are less trusted than itself.	APPNET0023	Security Permission (Allow Domain Policy Control)	CAT II
3.4.9.23	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="ControlEvidence"* (*Allow Evidence Control*) to code that is not highly trusted (has a strong name with a public key associated with a trusted entity) and has not been developed using secure coding techniques.	APPNET0024	Security Permission (Allow Evidence Control)	CAT II
3.4.9.24	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Security* permission is granted with *Flags="Assertion"* (*Assert any Permission that has been granted*) to code that is not a highly trusted extension to the CLR base libraries.	APPNET0025	Security Permission (Assert any Permission that Has Been Granted)	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**CAT**
3.4.9.25	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Performance Counter* permission is granted with *unrestricted="true"* to non-default permission set. The *Performance Counter* permission is granted to an unauthorized machine or category. The *Performance Counter* permission is granted with *Instrument* or *Administrator* access to code that does not provide or administer monitoring.	APPNET0026	Performance Counter Permission	CAT III
3.4.9.26	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Environment Variables* permission is granted with *unrestricted="true"* to a non-default permission set.	APPNET0027	Environment Variables Permission	CAT II
3.4.9.27	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Event Log* permission is granted with *unrestricted="true"* to a non-default permission set assigned to code that is not used to monitor system and application events.	APPNET0028	Event Log Permission	CAT II
3.4.9.28	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Registry* permission is granted with *unrestricted="true"* to a non-default permission set. *Registry* permissions are granted to unauthorized code.	APPNET0029	Registry Permission	CAT II
3.4.9.29	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Directory Services* permission is granted with *unrestricted="true"* (*Grant assemblies unrestricted access to all directory service paths* is selected) to a non-default permission set. The *Directory Services* permission specifies unauthorized service paths. *Write* access to the Windows system directory is granted to code that is not a trusted administrative tool. *Browse* access to the Windows system directory services (Active Directory/Global catalog, IIS Metabase) is granted to code that is not of local origin (with a strong name with a public key associated with a local entity).	APPNET0030	Directory Services Permission	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**CAT**
3.4.9.30	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	Strong names are simulated on a production system.	APPNET0031	Strong Name Membership Condition	CAT II
3.4.9.31	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	Non-default First Match Code Groups are defined.	APPNET0032	First Match Code Groups	CAT II
3.4.9.32	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	File Code Groups have been manually added to a .config file.	APPNET0033	File Code Groups, Net Code Groups	CAT II
3.4.9.33	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	Unauthorized code groups are assigned the *LevelFinal* attribute.	APPNET0035	Level Final Code Group Attribute	CAT III
3.4.9.34	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	A non-default code group is assigned the *Zone* Membership Condition.	APPNET0041	Zone Membership Condition	CAT II
3.4.9.35	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	CAS security is not enabled.	APPNET0045	Administering CAS Policy	CAT I
3.4.9.36	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Trust the Test Root* value is set to TRUE on a production system.	APPNET0046	Administering the Windows Environment for Test Root Certificates	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**CAT**
3.4.9.37	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Use expiration date on certificates* value is set to FALSE on a production system.	APPNET0047	Administering the Windows Environment for Expired Certificates	CAT II
3.4.9.38	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	An unauthorized code group is assigned the *Publisher* Membership Condition.	APPNET0048	Publisher Membership Condition	CAT II
3.4.9.39	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Check the revocation list* value is set to FALSE.	APPNET0049	Administering the Windows Environment for Revoked Certificates	CAT II
3.4.9.40	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Offline revocation server OK (Individual)* is set to TRUE. The *Offline revocation server OK (Commercial)* is set to TRUE. The *Java offline revocation server OK (Individual)* is set to TRUE. The *Java offline revocation server OK (Commercial)* is set to TRUE.	APPNET0050	Administering the Windows Environment for Unknown Certificate Status	CAT II
3.4.9.41	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *Invalidate version 1 signed objects* is set to FALSE. The *Check the revocation list on Time Stamp Signer* is set to FALSE.	APPNET0051	Administering the Windows Environment for Time Stamped Certificate Revocation	CAT II
3.4.9.42	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	The *StrongName* condition type listed for a non-default code group does not use a DOD PKI or an authorized third-party certificate.	APPNET0052	Strong Name Membership Condition	CAT II

Procedure Section #	Finding Information		Vulnerability Information		
Section #	**Status**	**Finding Details**	**SDID/ Vulnerability Key**	**Brief Description**	**CAT**
3.4.9.43	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	Non-default code group names are not unique.	APPNET0054	Administering CAS Policy for Group Names	CAT III
3.4.9.44	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	CAS policy and configuration files are not included as part of a reliable backup strategy.	APPNET0055	Administering CAS Policy and Policy Configuration File Backups	CAT II
3.4.9.45	☐ Finding ☐ Not a Finding ☐ Not Applicable ☐ Not Reviewed	Applications assigned the *typefilterlevel="Full"* attribute do not require authentication and encryption.	APPNET0060	Remoting Services Authentication and Encryption	CAT II

3. .NET FRAMEWORK SECURITY CHECKLIST AND PROCEDURES

This section details the procedures needed to perform a Security Readiness Review (SRR) of a .NET Framework installation.

3.1 Reviewer Notes

The .NET SRR is a manual process that uses the following tools: Microsoft .NET Framework Configuration Tool (MSCORCFG.MSC), CASPOL.EXE, SETREG.EXE, SN.EXE, and Microsoft Internet Explorer 6.0 or higher.

These tools are obtained in different ways for different versions of the .NET Framework.

Version 1: All of the listed tools with the exception of SN.EXE, SETREG.EXE and Microsoft Internet Explorer are installed with the .NET Framework. SN.EXE is included in the .NET Framework SDK installation. SETREG.EXE is included on Windows Server installations. Microsoft Internet Explorer can be obtained from http://www.microsoft.com.

Version 1.1: All of the listed tools with the exception of SN.EXE, SETREG.EXE and Microsoft Internet Explorer are installed with the .NET Framework. SN.EXE is included in the .NET Framework SDK installation. SETREG.EXE is included on Windows Server installations. Microsoft Internet Explorer can be obtained from http://www.microsoft.com.

Version 2.0: All of the listed tools with the exception of SETREG.EXE and Microsoft Internet Explorer are included in the .NET Framework SDK installation. SETREG.EXE is included on Windows Server installations. Microsoft Internet Explorer can be obtained from http://www.microsoft.com.

The .NET Framework executables can typically be found in the %systemroot%\Microsoft.NET\Framework\<version>\ directory.

The .NET Framework SDK executables will be located in the installation directory that was chosen when the SDK was installed.

The following instructions indicate some details specific to version 1.1 of the .NET Framework. Where the version being reviewed differs from version 1.1, the reviewer should use a corresponding or similar selection appropriate for the version under review.

3.2 IAVM Compliance

IAVM alerts, bulletins, and advisories were instituted to provide positive control of vulnerability notification and corresponding corrective action within DOD. All DOD program managers and system administrators, and/or other personnel responsible for system networks shall comply with the IAVM process. Security patches that address .NET vulnerabilities are reviewed during an operating system security review and are not included in this checklist.

3.3 Determining Which Versions of the .NET Framework Are Installed

To determine which versions of the .NET Framework are installed perform the following steps:

1. Open Windows Explorer.
2. Navigate to the %SystemRoot%\Microsoft.NET\Framework Folder.
3. The Frameworks will be installed in the following directories. Note: If any other directories exist then this may indicate that a beta version of the framework is installed.
   - v1.0.3705
   - v1.1.4322
   - v2.0.50727
4. Inside each of the listed folders (if it exists) locate the Mscorlib.dll file, right click on the file, and select *Properties* and then the *Version* tab. Consult the following table to determine the version installed. (If the mscorlib.dll file does not exist then that version of the framework is not installed).

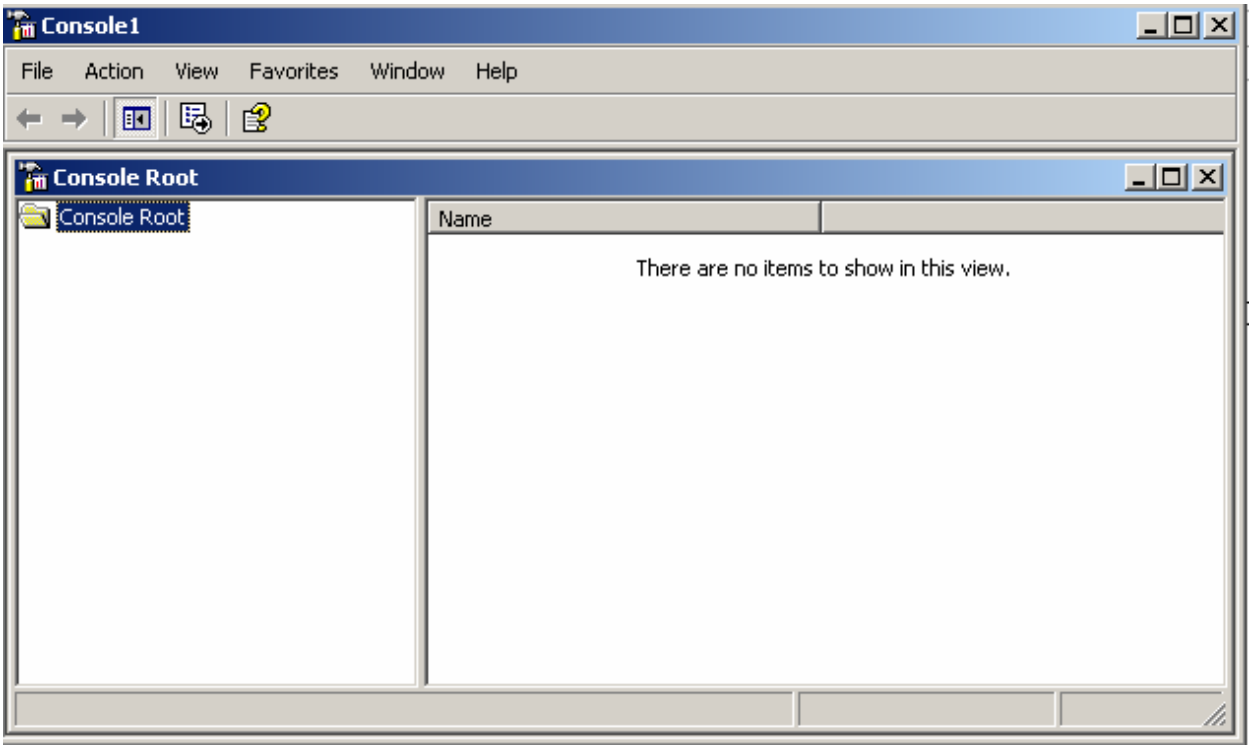
File Version	Framework Version
1.0.3705.0	1.0 RTM
1.0.3705.209	1.0 SP1
1.0.3705.288	1.0 SP2
1.0.3705.6018	1.0 SP3
1.1.4322.573	1.1 RTM
1.1.4322.2032	1.1 SP1 on Windows XP
1.1.4322.2300	1.1 SP1
2.0.50727.42	2.0 RTM

3.4 Reviewer Interfaces

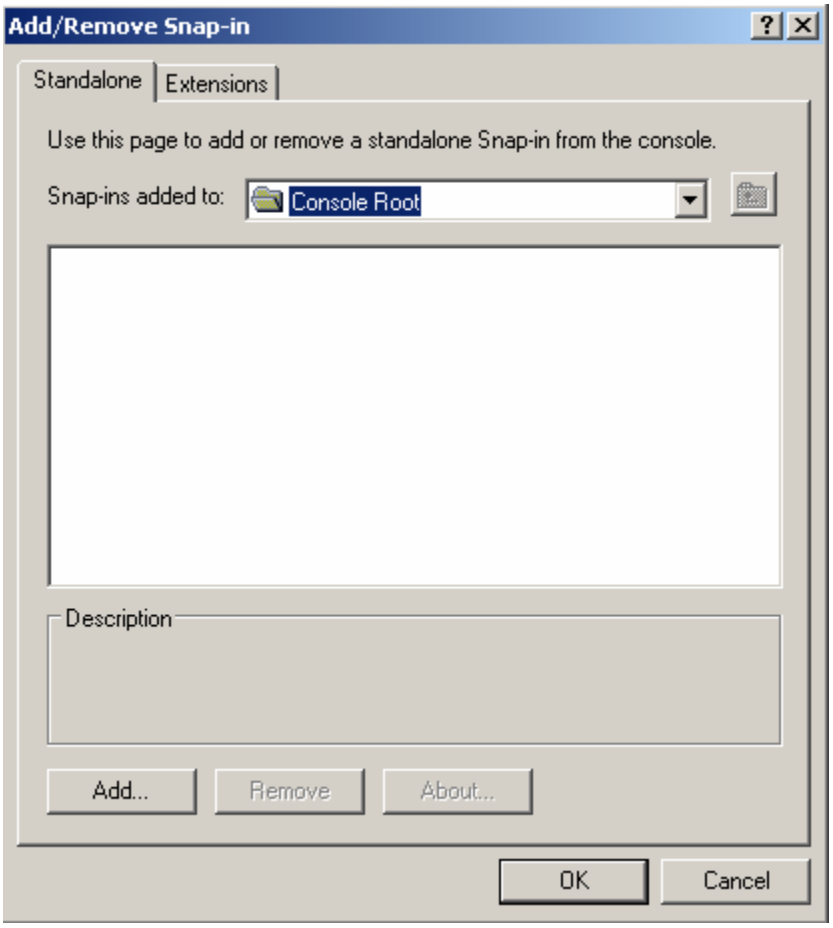
3.4.1 Using the Microsoft .NET Framework Configuration Tool MSCORCFG.MSC

The .NET Framework configuration tool, MSCORCFG.MSC, is a graphical user interface (GUI) tool used to configure and view the .NET Framework. The configuration available through this tool includes some security elements.

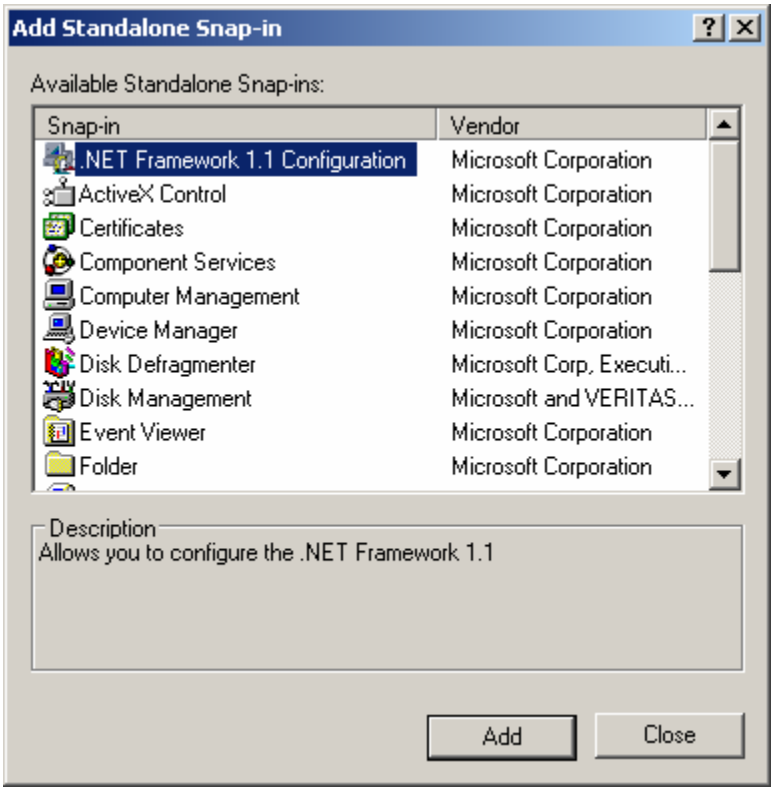
1) Open the Microsoft Management Console (MMC) by Selecting *Start->Run*, type MMC in the Open combo box and then press <ENTER>.



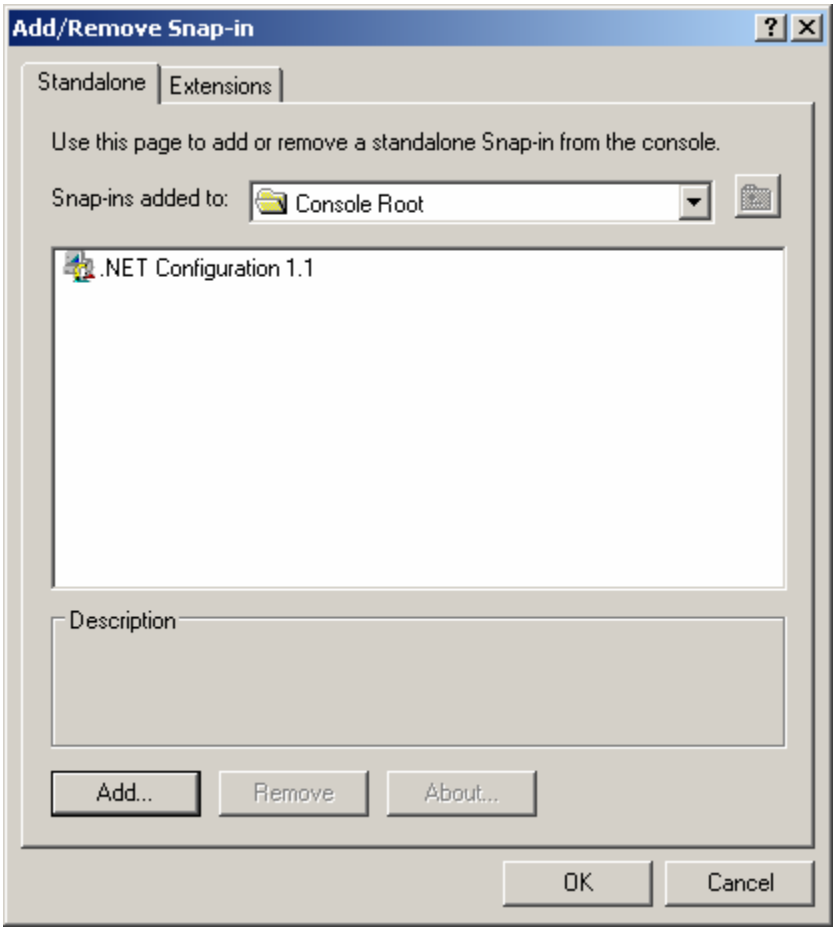
2) Select *File->Add/Remove Snap In…* from the main menu.



3) Select the *Add* button.

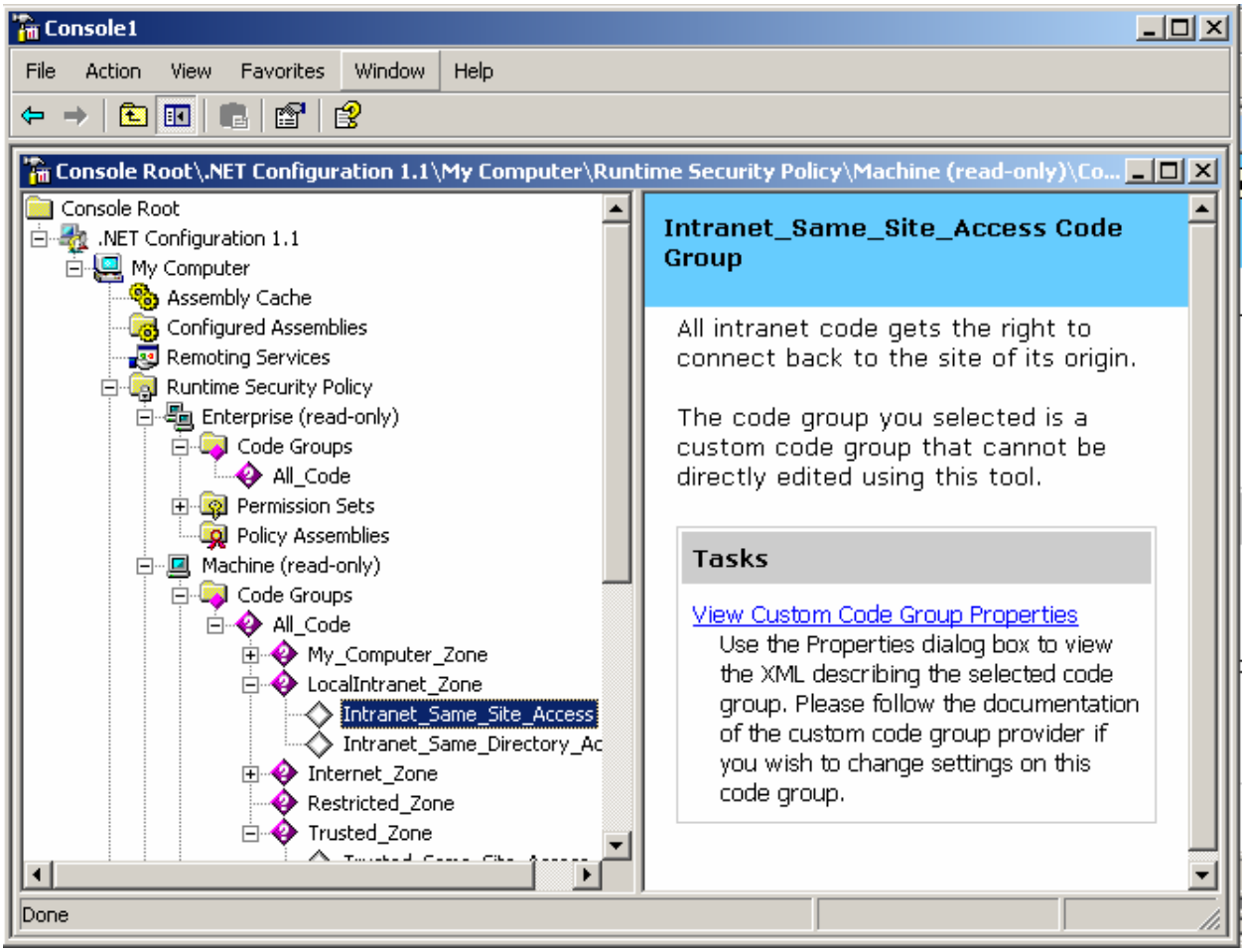


4) Select the *Microsoft .NET Framework X.X Configuration* from the list then press the *Add* button.
5) Select the *OK* button.



6) The appropriate configuration tool has now been added to the MMC.

7) Expand the *.NET Configuration X.X* tree branch, then expand the *My Computer* branch. The following entries will be displayed under *My Computer: Assembly Cache*, *Configured Assemblies*, *Remoting Services*, *Runtime Security Policy*, and *Applications*.
8) Expand the *Runtime Security Policy*. Three base policy levels will be displayed (*Enterprise*, *Machine*, and *User*).
9) Expand *Enterprise* to display the policy components. These same policy components are also found under the *Machine* and *User* policy levels.
10) Expand *Machine*.
11) Expand *User*.
12) Expand all the *Code Groups*, *Child Code Groups*, and *Permission Sets*.



3.4.2 Using the .NET Framework Code Access Security (CAS) Policy Tool CASPOL.EXE

The caspol.exe tool may be used to dump .NET Framework code access security configuration. Each version of the .NET Framework comes with its own version of caspol.exe. Each version of caspol.exe can only be used to administer the .NET Framework version for which it was built. Use the version of CASPOL.EXE found in the same directory structure as the .NET Framework version that is being reviewed.

Following are caspol.exe command line syntaxes for displaying the .NET code groups, permission sets, and trust assemblies. Issue all commands from the Windows command line accessed via the Windows Start>>Run>> open: cmd. Output may be directed to a text file with the use of the > redirection pipe. The resulting file may then be reviewed using the Windows Notepad or other text editor.

To list code groups for all levels, type the following command at the command prompt. Text within "[]" indicates a command line option. The "[]" characters are not part of the command line option and should not be included.

caspol.exe –all –lg [> c:\temp\AllGroups.txt]

To list permission sets for all levels, type the following command.

caspol.exe –all –lp [> c:\temp\PermissionSets.txt]

To list full trust assemblies for all levels, use the following command.

caspol.exe –all –lf [> c:\temp\AllTrustAssemblies.txt]

3.4.3 Using the Strong Name Tool SN.EXE

The Strong Name tool, sn.exe, is a command line tool that is used to display and configure strong names for application code. It is supplied with the .NET Framework SDK and is not available on systems where the SDK has not been installed. Issue all commands from the Windows command line accessed via the Windows Start>>Run>> open: cmd. Use the command line listed below to gather information about strong names required for the review. Output may be directed to a text file with the use of the > redirection command. The resulting file may then be reviewed using the Windows Notepad or other text editor.

sn.exe –V1 [>c:\temp\SNSimulation.txt]

3.4.4 Using the Software Publishing State Tool SETREG.EXE

The Software Publishing State tool, setreg.exe, is a command line tool that is used to display and configure the Software Publishing State registry keys for application code. It is supplied with the .NET Framework SDK and most server versions of the Windows operating system. Issue the setreg command from the Windows command line accessed via the Windows Start>>Run>> open: cmd. Use the command line listed below to gather information about the software publishing state for the review. Output may be directed to a text file with the use of the > redirection command. The resulting file may then be reviewed using the Windows Notepad or other text editor.

Setreg.exe [>c:\temp\SetReg.txt]

3.4.5 Review Results

The reviewer records finding results from the checks, in the Section 2 Results List. Results may then be recorded in VMS if the system is a registered VMS asset.

3.4.6 Version-specific Vulnerabilities

Vulnerabilities that apply only to a specific version of .NET are so noted. Versions to which the vulnerability does not apply should have that finding marked as N/A.

3.4.7 Assemblies, Evidence, Permission Sets, and Code Groups

Programs written for the .NET Framework execute with the credentials of the user account used to launch the program. As such these programs are constrained by any operating system security settings that may be in place. This is identical to the operation of any non .NET application. The additional restrictions that may be imposed through the .NET Framework are designed to further restrict .NET applications, providing an additional layer of protection.

The following components are used to establish which permissions are granted by the .NET Framework.

Assemblies – an assembly is the .NET Frameworks' term for a program. An assembly may consist of multiple executables, DLLs, and libraries.

Evidence – Evidence is information about an assembly. Evidence may be contained in the assembly itself or may be presented by the host. There are currently seven types of evidence in the .NET Framework. These evidence types are:

- Application Directory – the directory where the assembly resides.
- Hash – a cryptographic hash of the assembly.
- Publisher – The publisher of the application, based upon Authenticode signing of the assembly.
- Site – The site where the assembly originated. This is only valid when the assembly is executed directly from the site.
- StrongName – A cryptographic signing of the assembly.
- URL – The URL where the assembly originated. This is only valid when the assembly is executed directly from the URL.
- Zone – The Internet Explorer Security Zone associated with the site of origin for the assembly.

Permission Sets – Permission Sets are groups of permission that can be granted to .NET Assemblies. There are several default Permission Sets, and non-default Permission Sets may be created.

Code Group – A code group is used to assign a Permission Set to an Assembly. Assemblies are placed into 1 or more Code Groups based upon the Evidence they present. As part of membership detection any membership conditions for parent code groups must also be met.

When performing an SRR of the .NET Framework it is not enough to simply evaluate the permissions assigned to a Permission Set to determine whether a vulnerability exists or not. Code Groups which are granted that permission set must be considered as part of the evaluation process to ensure that potentially dangerous permissions are not granted to unapproved assemblies.

For example: Access to the file system is one of the permissions that can be granted through a Permission Set. These permissions range from no access to the file system, to limited access to specific files or directories, to full access to the file system. A Permission Set that grants unrestricted access to the file system is not a vulnerability in and of itself. However if that Permission Set were granted to a Code Group whose membership condition was the Internet Zone , essentially granting full file system access to any program downloaded from the Internet, then this would be a vulnerability. The same Permission Set assigned to a Code Group whose membership is restricted by a Strong Name signed assembly, where the keys used for the signing are controlled by the site, would not be considered a vulnerability.

Note: In the example above, an Assembly that is granted unrestricted access to the file system would still be restricted by the File ACLS of the system.

Non-default Code Group – A code group not part of the default installation of .Net.

Default Code Groups – Code groups installed by the default installations of .Net 1.0, 1.1 & 2.0.

Enterprise Policy:

All_Code

Membership Condition: All Code
Permission Set: Full Trust

Machine Policy:

All_Code

Membership Condition: All Code
Permission Set: Full Trust

My_Computer_Zone

Membership Condition: Zone
Zone: My Computer
Permission Set: Full Trust

Microsoft_Strong_Name

Membership Condition: Strong Name
Public key:

00240000048000009400000006020000002400005253413100040000010001007D1FA57C4AED9F0A32E84AA0FAEFD0DE9E8FD6AEC8F87FB03766C834C99921EB23BE79AD9D5DCC1DD9AD236132102900B723CF980957FC4E177108FC607774F29E8320E92EA05ECE4E821C0A5EFE8F1645C4C0C93C1AB99285D622CAA652C1DFAD63D745D6F2DE5F17E5EAF0FC4963D261C8A12436518206DC093344D5AD293

Permission Set: Full Trust

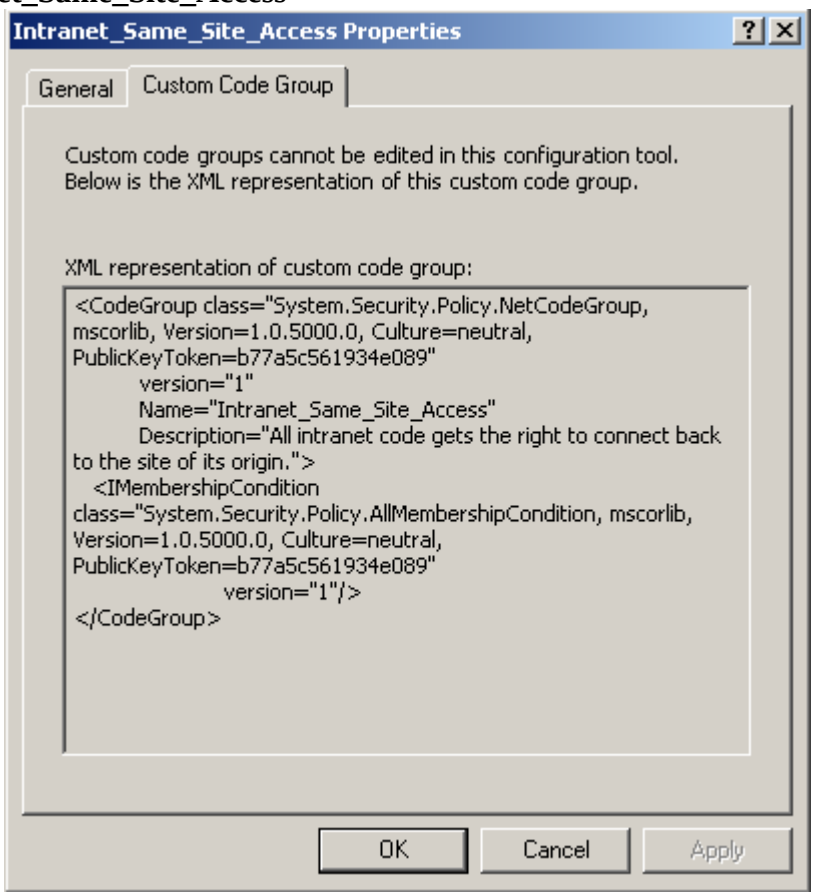
ECMA_Strong_Name

Membership Condition: Strong Name
Public key: 0000000000000000040000000000000000
Permission Set: Full Trust

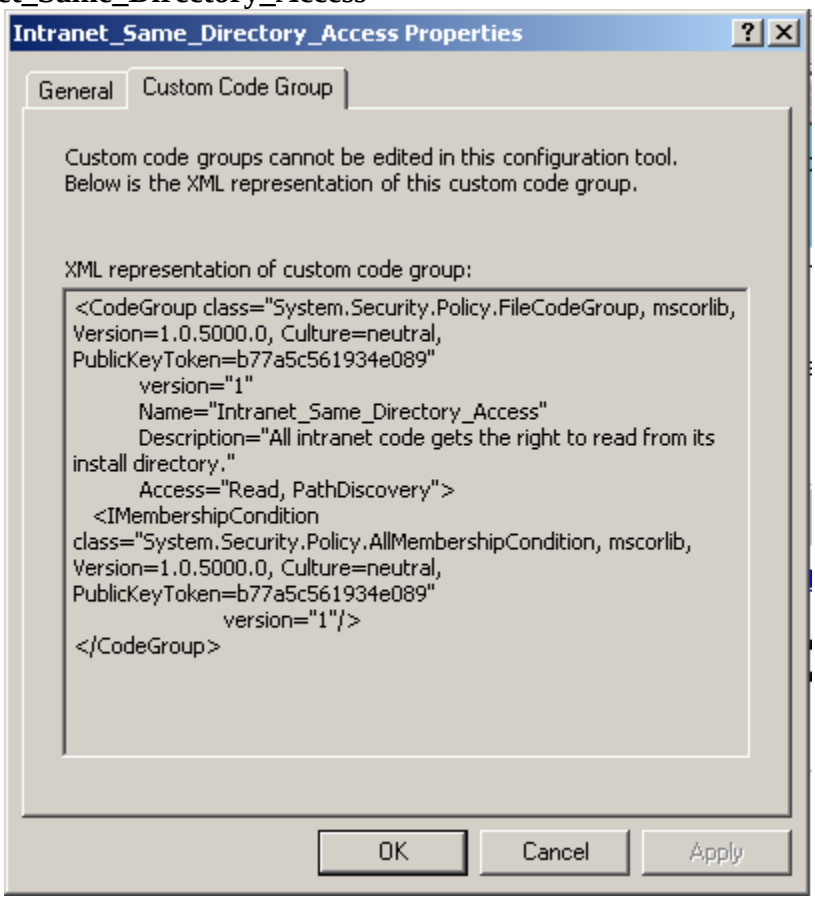
LocalIntranet_Zone

Membership Condition: Zone
Zone: Local Internet
Permission Set: LocalInternet

Intranet_Same_Site_Access



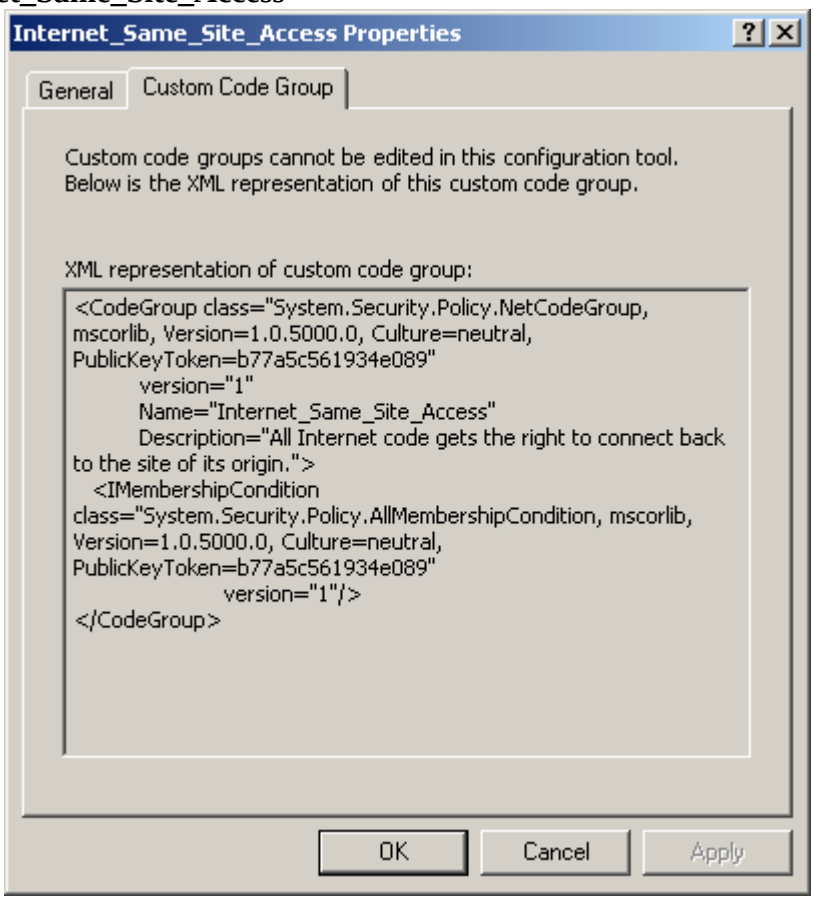
Intranet_Same_Directory_Access



Internet_Zone

Membership Condition: Zone
Zone: Internet
Permission Set: Internet

Internet_Same_Site_Access



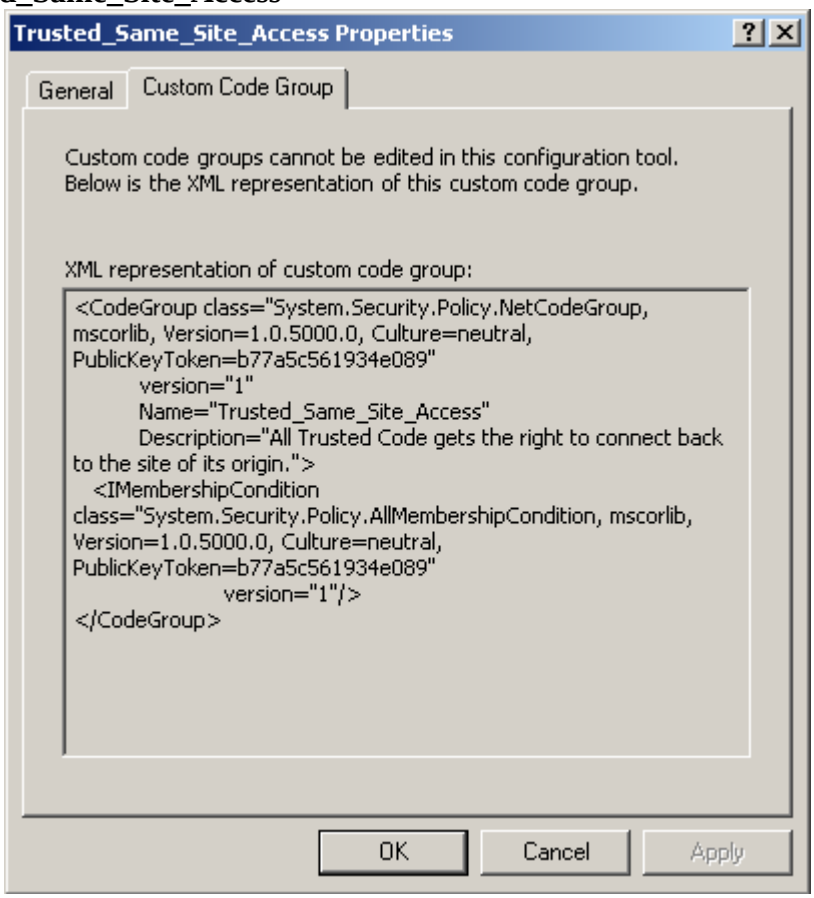
Restricted_Zone

Membership Condition: Zone
Zone: Untrusted Sites
Permission Set: Nothing

Trusted_Zone

Membership Condition: Zone
Zone: Trusted Sites
Permission Set: Internet

Trusted_Same_Site_Access



User Policy:

All_Code

Membership Condition: All Code
Permission Set: Full Trust

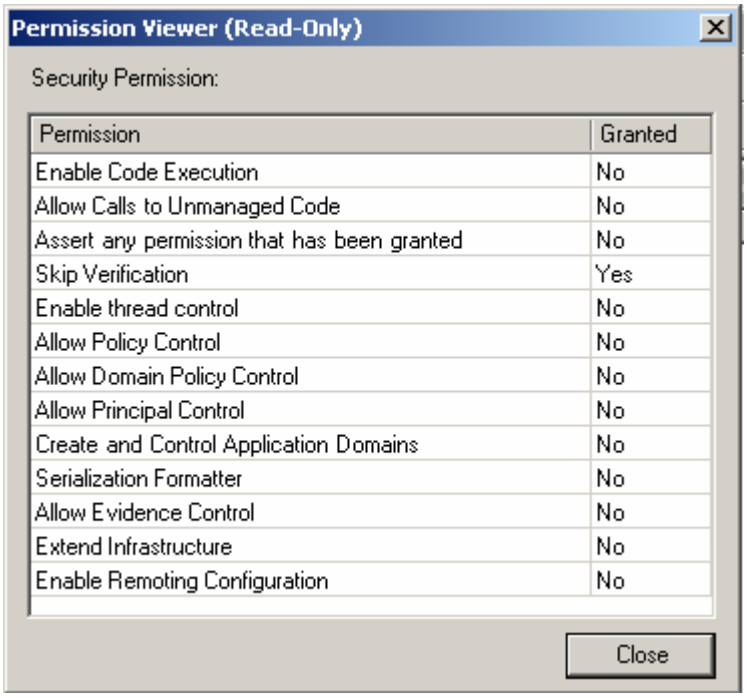
All other code groups are considered non-default code groups and must be checked for security compliance. If the default code groups have been modified then they should be considered a non-default code group for review purposes.

Non-default Permission Set – A permission set not part of the default installation of .Net. Default Permission Sets are as follows:

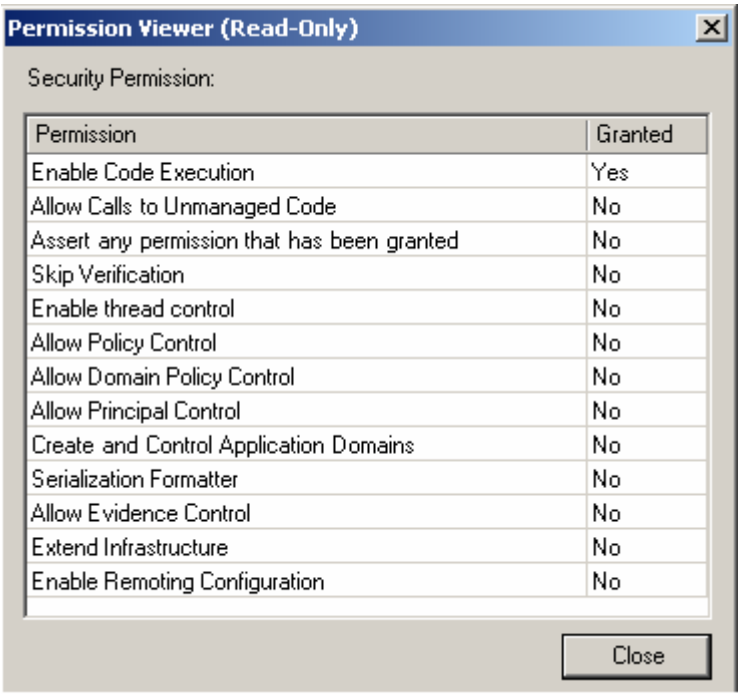
The default permission sets are the same for each policy level (User, Machine, Enterprise)

1) FullTrust - This permission set gives code unrestricted access to all protected resources.

2) SkipVerification - This permission set gives code the Security Permission with Skip Verification granted.

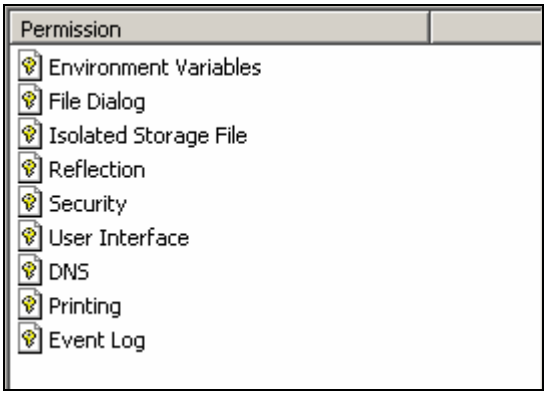


3) Execution - This permission set gives code the Security Permission with Skip Verification granted.

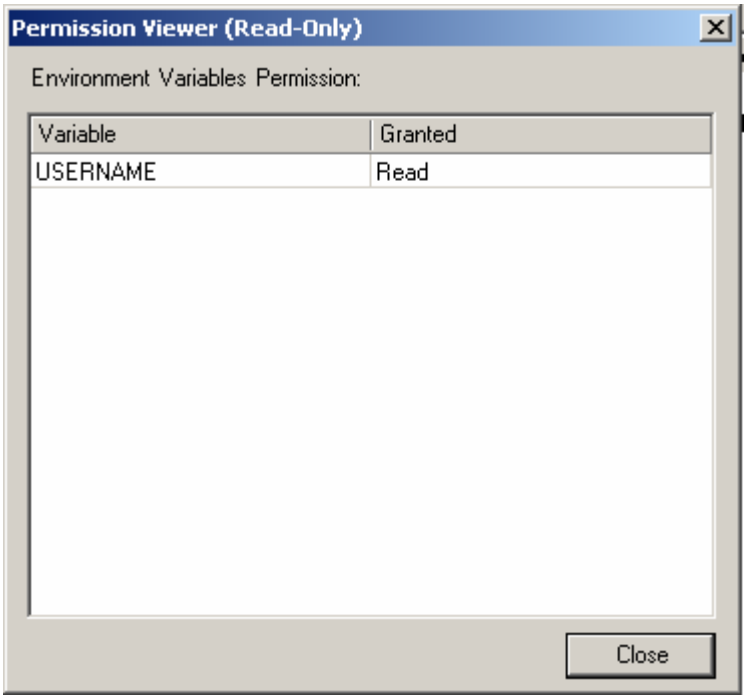


4) Nothing - This permission set grants no permissions to code.

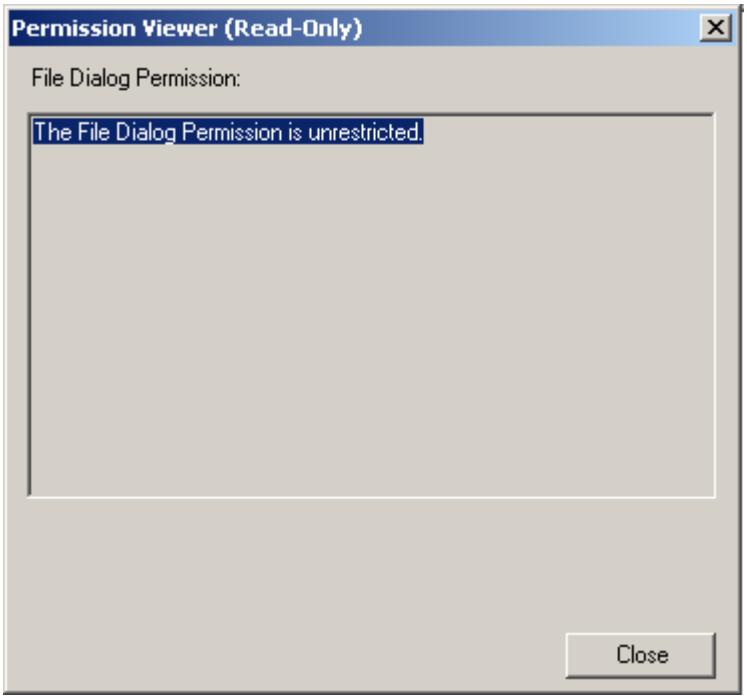
5) LocalIntranet - This permission set includes the permissions displayed below



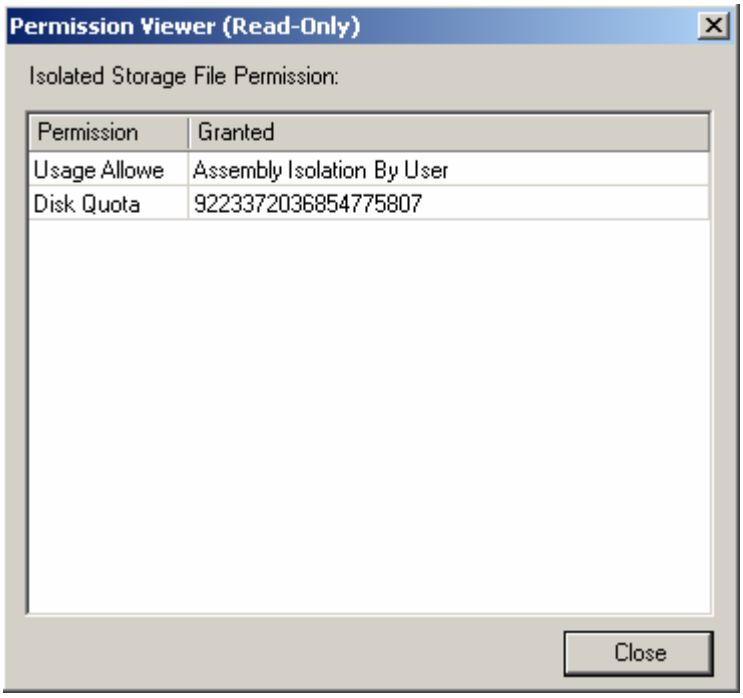
5.1) Environment Variables



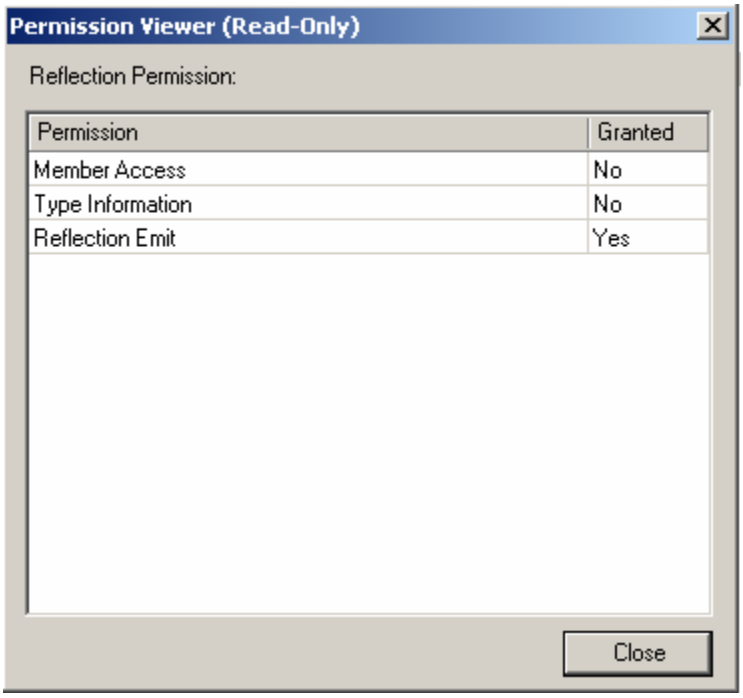
5.2) File Dialog



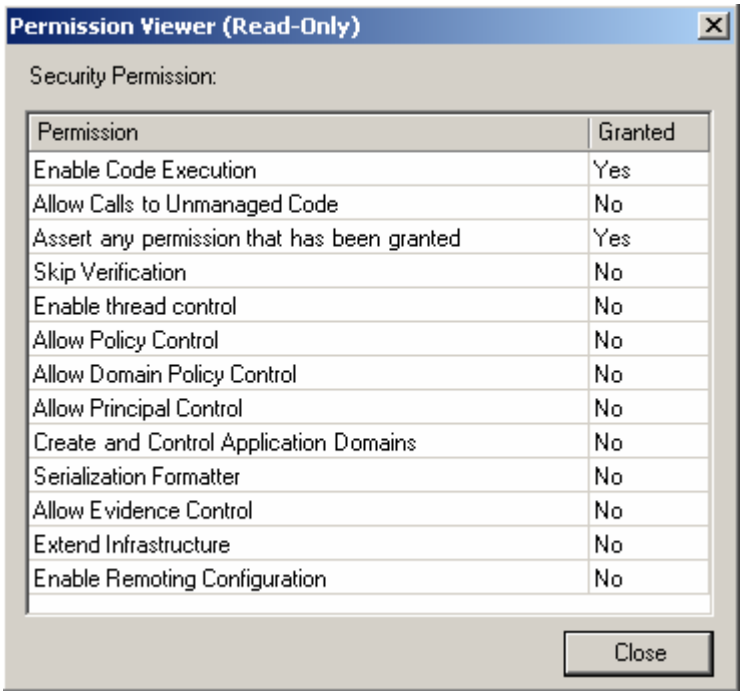
5.3) Isolated File Storage



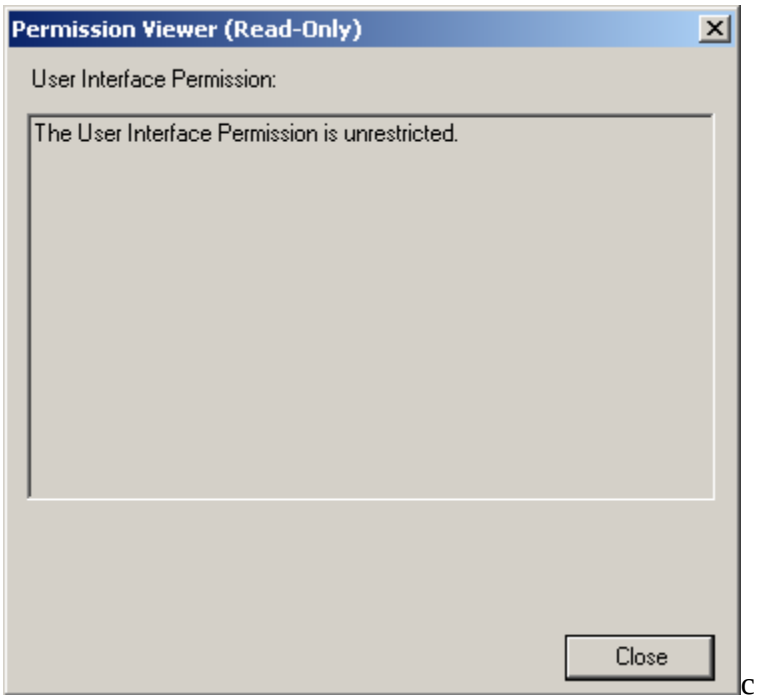
5.4) Reflection



5.5) Security

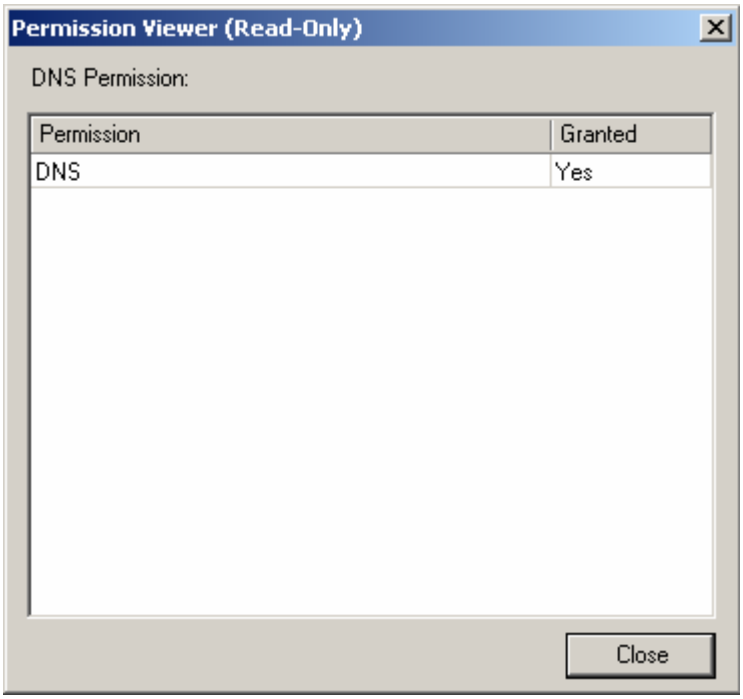


5.6) User Interface

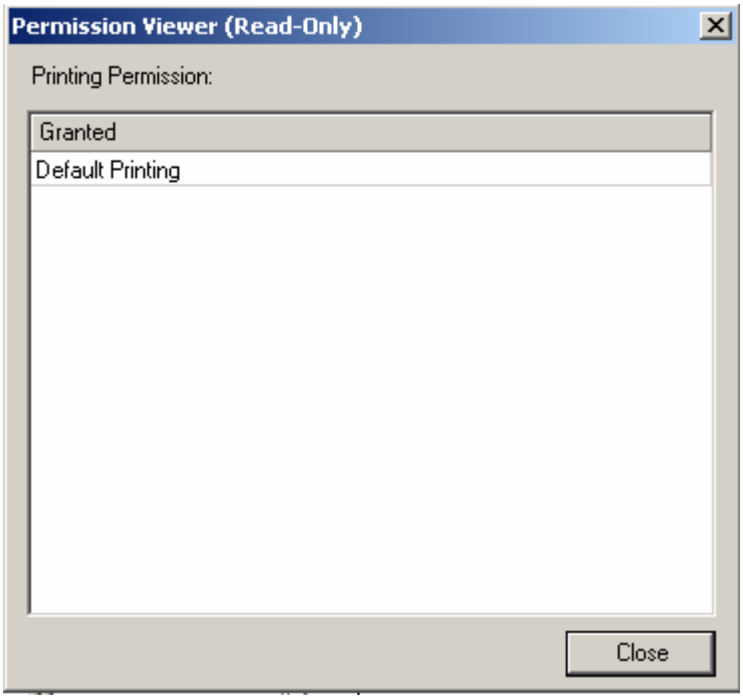


C

5.7 DNS

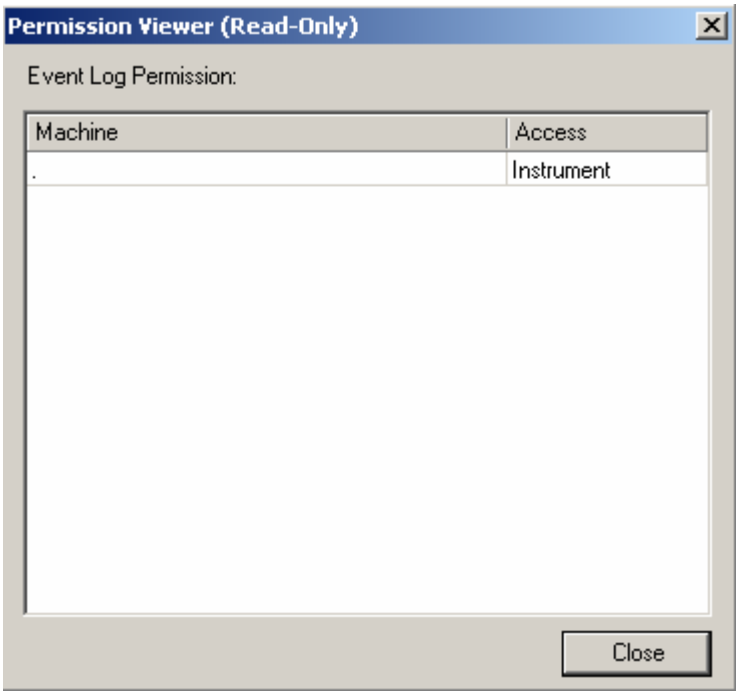


5.8) Printing

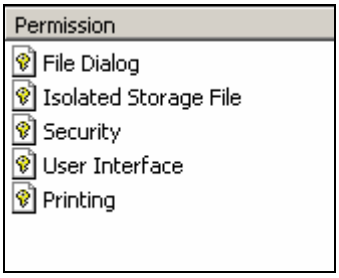


5.9) Event Log

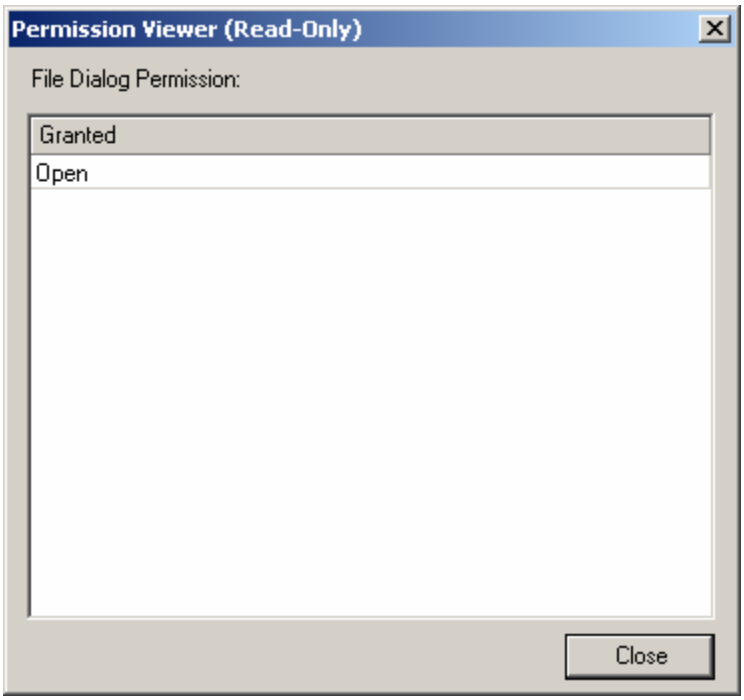
The Event Log Permissions only applies to .Net 1.0 and .Net 1.1.
The Event log permission is not included with .Net 2.0.



6) Internet – This permission set includes the permissions displayed below

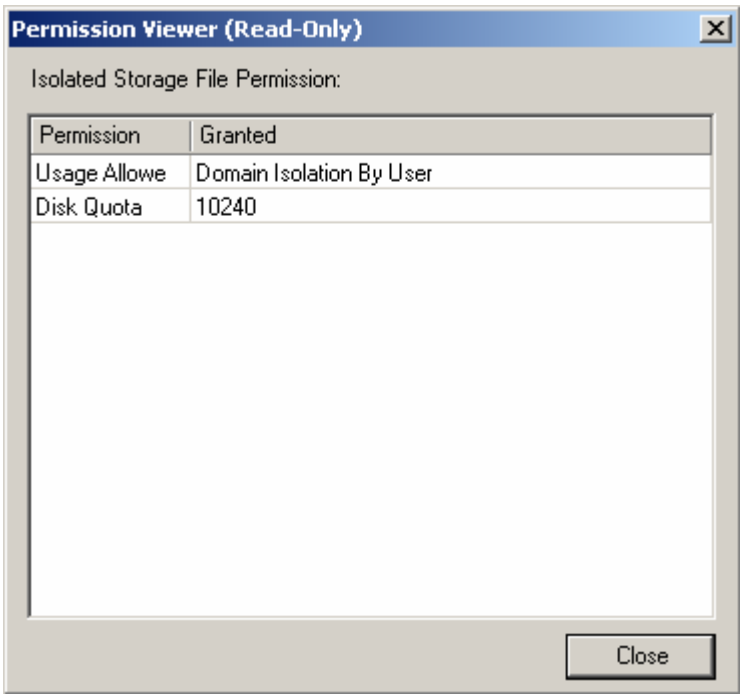


6.1) File Dialog

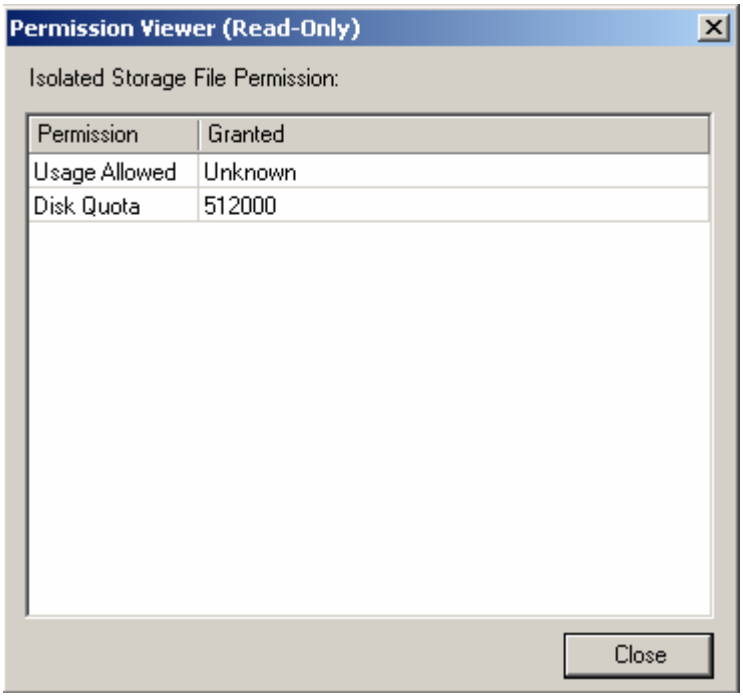


6.2) Isolated Storage File

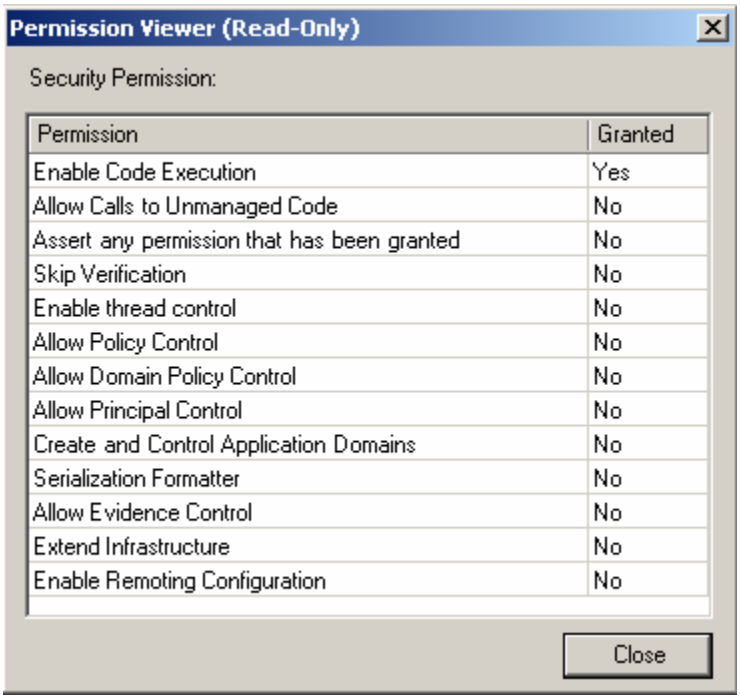
.Net 1.0 and .Net 1.1 Default



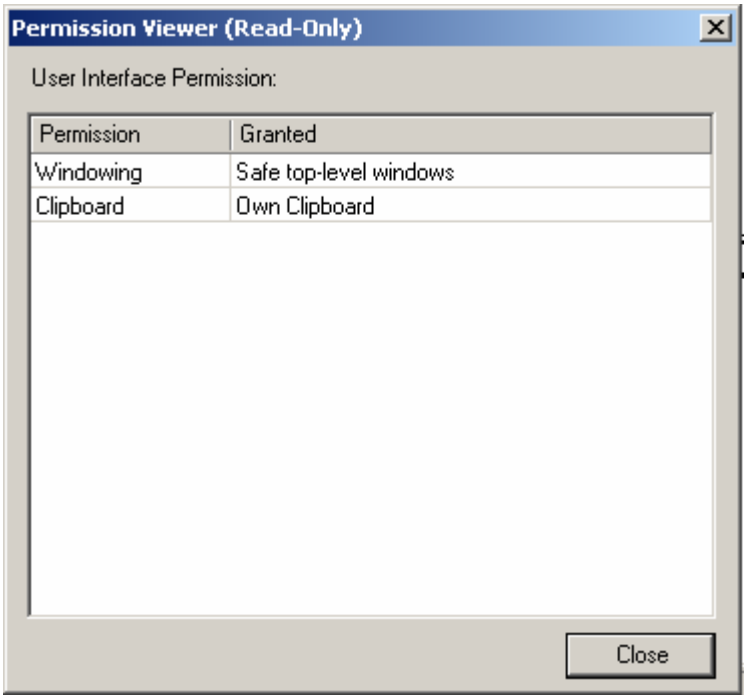
.Net 2.0 Default



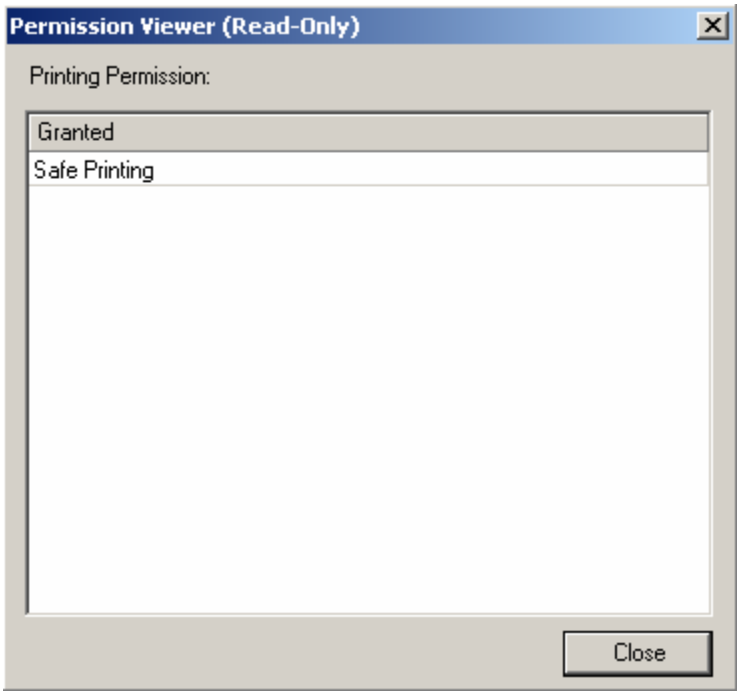
6.3) Security


Permission Viewer (Read-Only)
Security Permission:
Permission
Granted
Enable Code Execution
Yes
Allow Calls to Unmanaged Code
No
Assert any permission that has been granted
No
Skip Verification
No
Enable thread control
No
Allow Policy Control
No
Allow Domain Policy Control
No
Allow Principal Control
No
Create and Control Application Domains
No
Serialization Formatter
No
Allow Evidence Control
No
Extend Infrastructure
No
Enable Remoting Configuration
No
Close

6.4) User Interface

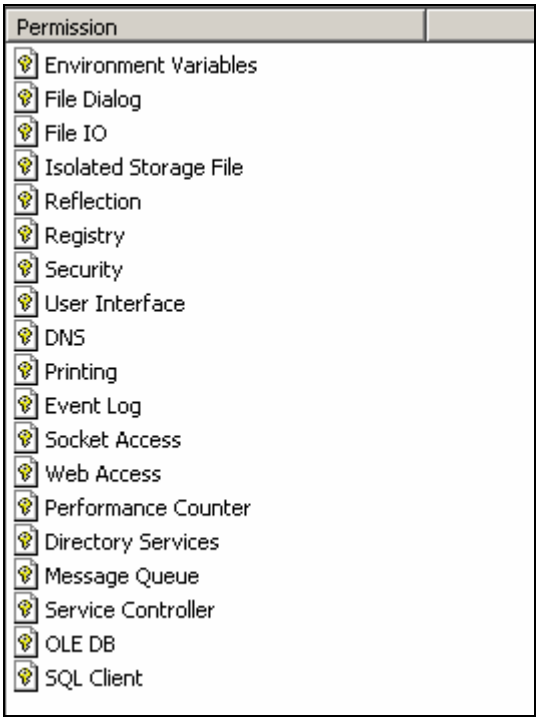


6.5) Printing

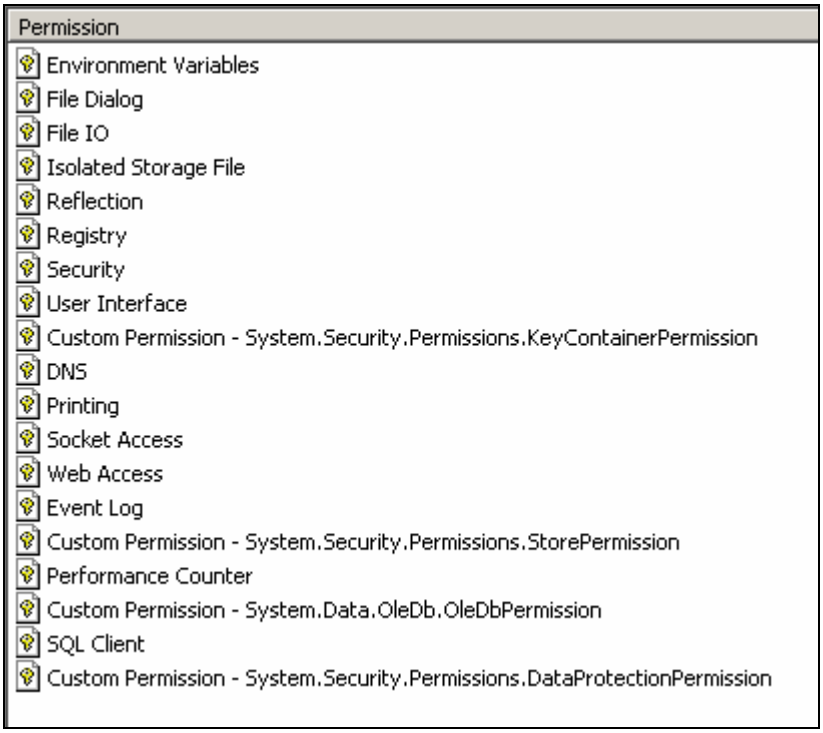


7) Everything

.Net 1.0 and .Net 1.1 default

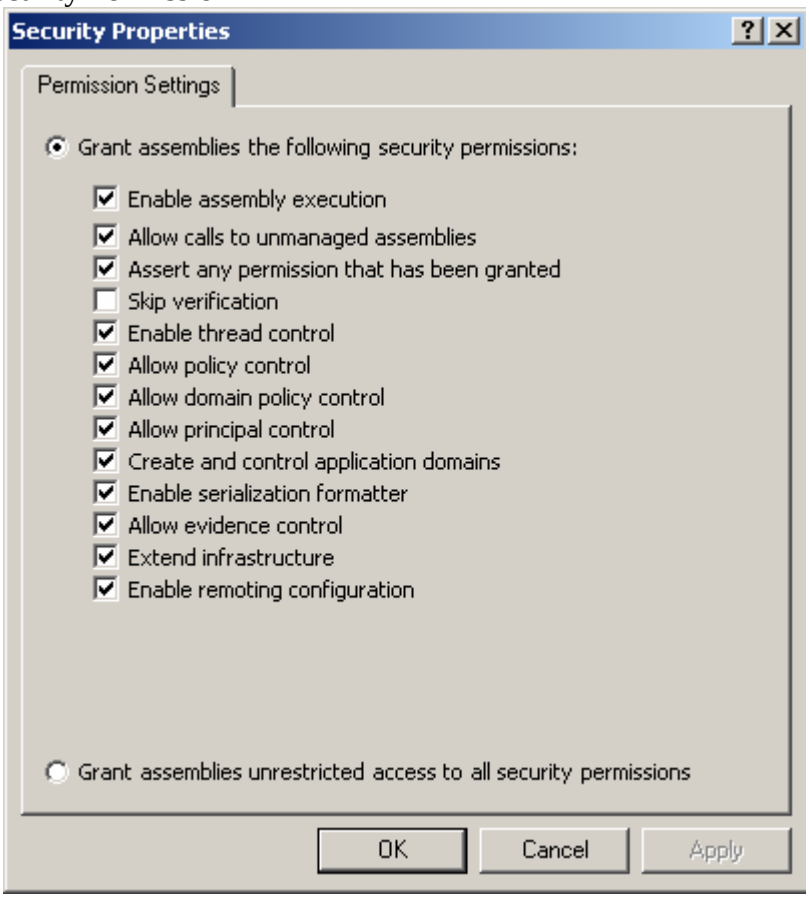


.Net 2.0



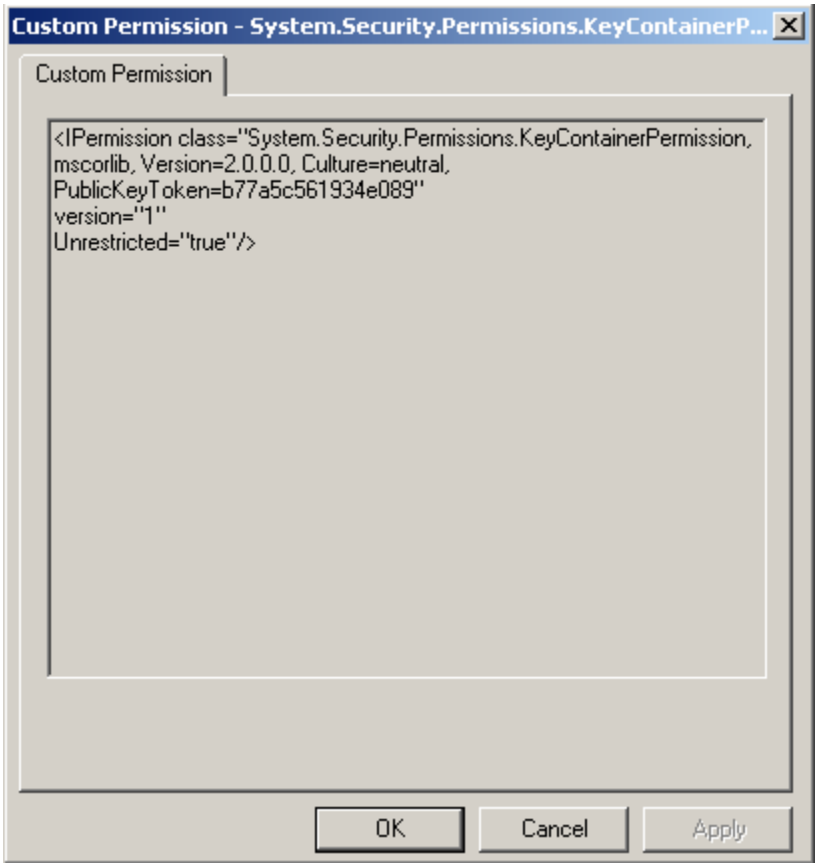
The permissions displayed above all grant code unrestricted access with the following exceptions noted below.

7.1) Security Permission

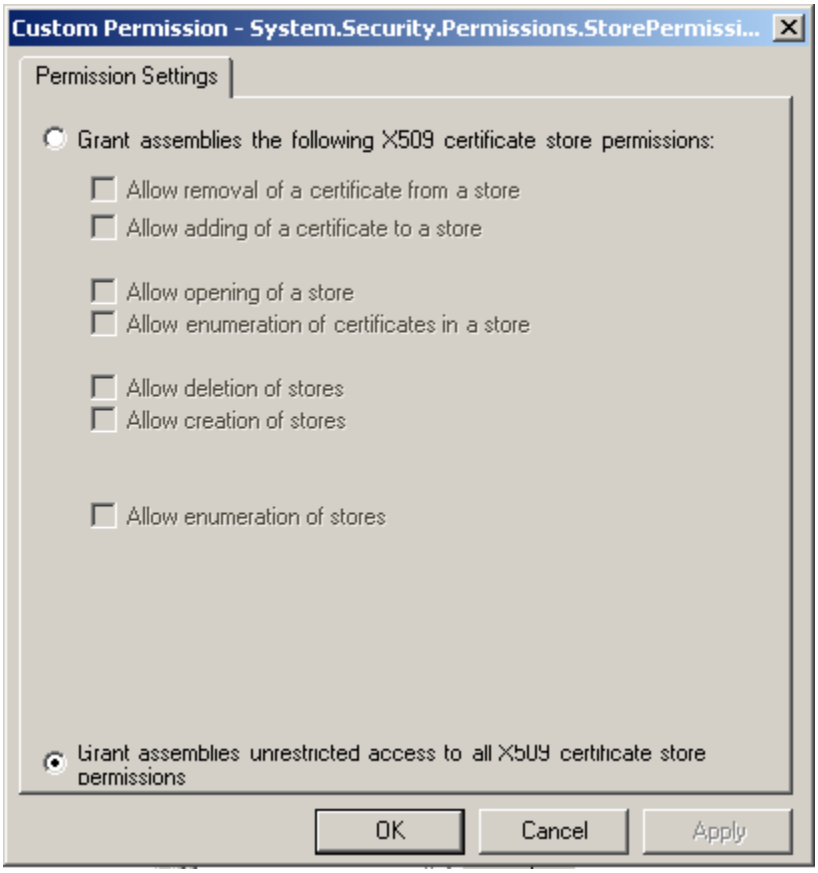


.Net 2.0 Everything Permission set contains 4 additional security permissions not included in .Net 1.0 and .Net 1.1

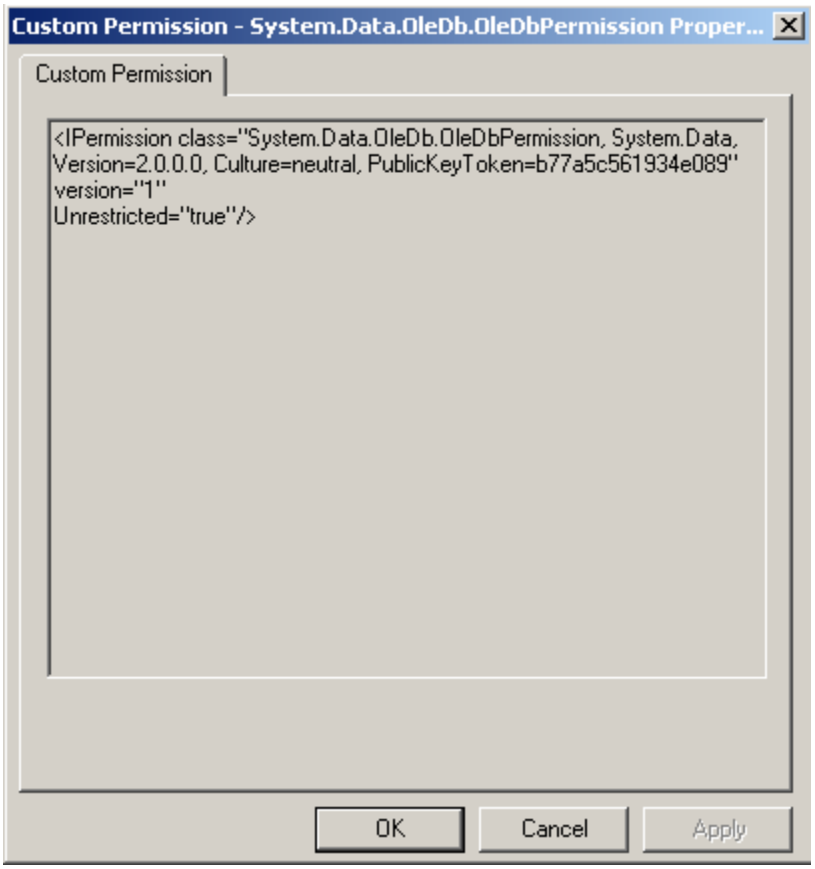
7.2) Custom Permission - System.Security.Permissions.KeyContainerPermission



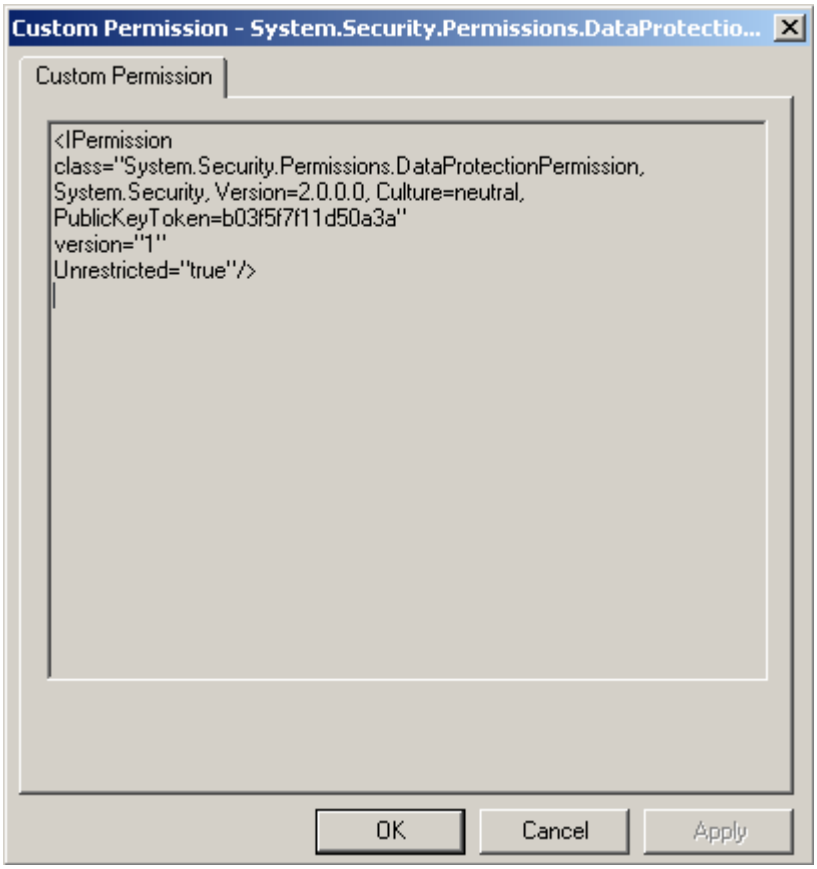
7.3) Custom Permission – System.Security.Permissions.StorePermission



7.4) Custom Permission - System.Data.OleDb.OleDbPermission



7.5) Custom Permission - System.Security.Permissions.DataProtectionPermission



All other permission sets are considered to be non-default permission sets and code groups using these permissions should be treated as non-default code groups.

3.4.8 Determining Effective Permissions

.NET Framework security policies can be defined at four levels: Enterprise, Machine, User, and Application Domain. Of these four, only the Enterprise, Machine, and User levels will be considered as part of the evaluation process. The configuration information for each level is stored in configuration files within each .NET Framework directory. There is currently no central management capability, so these files must be copied to every system in order for them to be effective.

Level	File
Enterprise	%SystemRoot%\Microsoft.NET\Framework\vX.X.XXXX\enterprisesec.config
Machine	%SystemRoot%\Microsoft.NET\Framework\vX.X.XXXX\security.config
User	%SystemRoot%\Microsoft.NET\Framework\vX.X.XXXX\user.config

Table 1 - Location of Configuration Files

Determining the effective permissions for a given assembly involves determining which code groups the assemblies belong to and combining all of the permission sets granted to the assembly to arrive at an effective permission set. Please refer to the *NSA Guide to Microsoft .NET Framework Security*, pages 66 – 71 for a detailed description of the rules and procedures used to determine the effective permission set.

3.4.9 .NET Security Framework Checks and Procedures

The following instructions should be used for checks APPNET0001 through APPNET0030. Use the permission name specified in the individual instruction to determine the specific permission to review. Any instructions included under a check are specific to that check and should be followed.

Performing a SRR on the .NET Framework involves identifying which permission sets have potentially dangerous permissions. After determining which permission sets are of interest the code groups to which those permission sets are assigned must be identified. Once the code groups are identified the membership conditions for those code groups must be documented. When determining whether or not a given vulnerability exists the reviewer must evaluate the membership conditions of code groups that grant the permission and determine if the permission is restricted to the appropriate assemblies. When entering the finding details the reviewer should include the name of the permission set(s) and the code group(s) that grant the permission set(s).

CASPOL.EXE: Review the caspol.exe listing for all permission sets. Search for all instances of the permission name specified, making note of the permission properties. Permission set names are numbered and precede the list of permissions assigned.

MSCORCFG.MSC: For each policy level (*Enterprise, Machine, User)*, review the permissions assigned to all permission sets. One-by-one, select a permission set in the left-hand frame to review. If the permission is listed in the right hand frame, then right-click on it and select *Properties* or select *View Permissions* if the *Help topic* is displayed. View any radio button selected or, if properties are displayed in a table, view any values listed in associated fields.

Security Permissions: The Security permission includes a list of possible individually assigned sub-permissions. This listed is presented in table format in the MSCORCFG.MSC application. Note any checkboxes selected when *Grant assemblies the following security permissions:* is selected. In CASPOL.EXE, individual security permissions are listed individually under separate permissions type *Ipermission class="System.Security.Permissions. SecurityPermission…)* and indicated by name with the *Flags=* specifier.

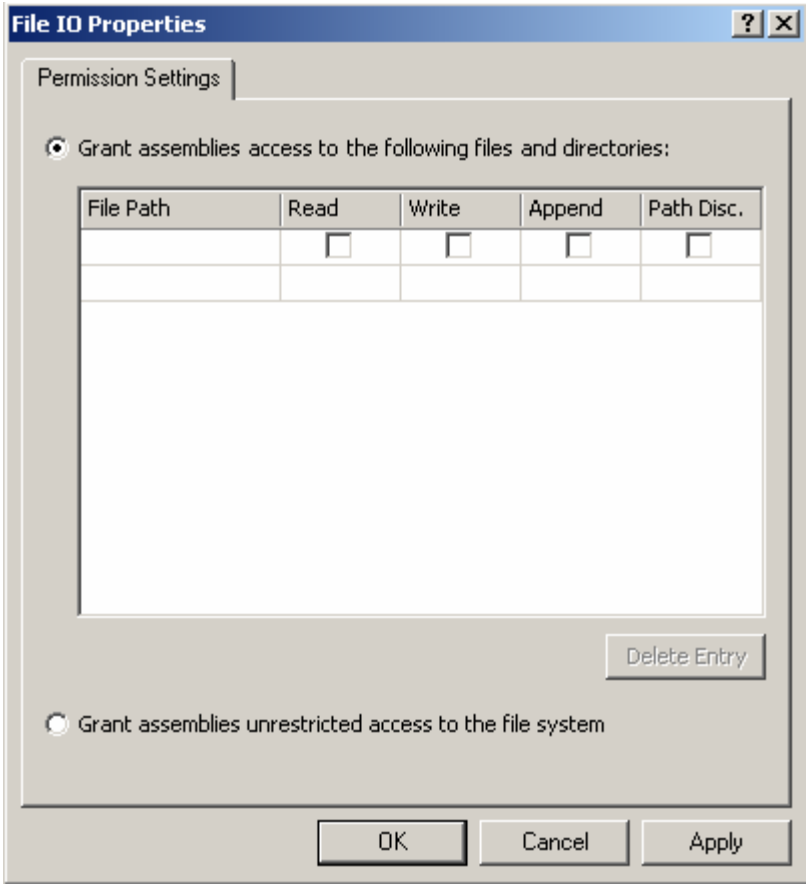
3.4.9.1 APPNET0001: File IO Permission

Description: The *File IO* permission allows an application to access system files directly.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *FileIOPermission.*

MSCORCFG.MSC: If *Grant assemblies access to the following files and directories* is selected, note the file path(s) entered and which permissions are granted.



Validate:

1. If a Permission Set with the *File IO* permission of *Grant Assemblies unrestricted access to the file system (unrestricted="true")*is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership

condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.

2. If a Permission Set granting limited *File IO* permissions is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0001	File IO Permission	
Reference:	.NET Framework Security Guide pg. 14	IA Control: ECCD-1, ECCD-2

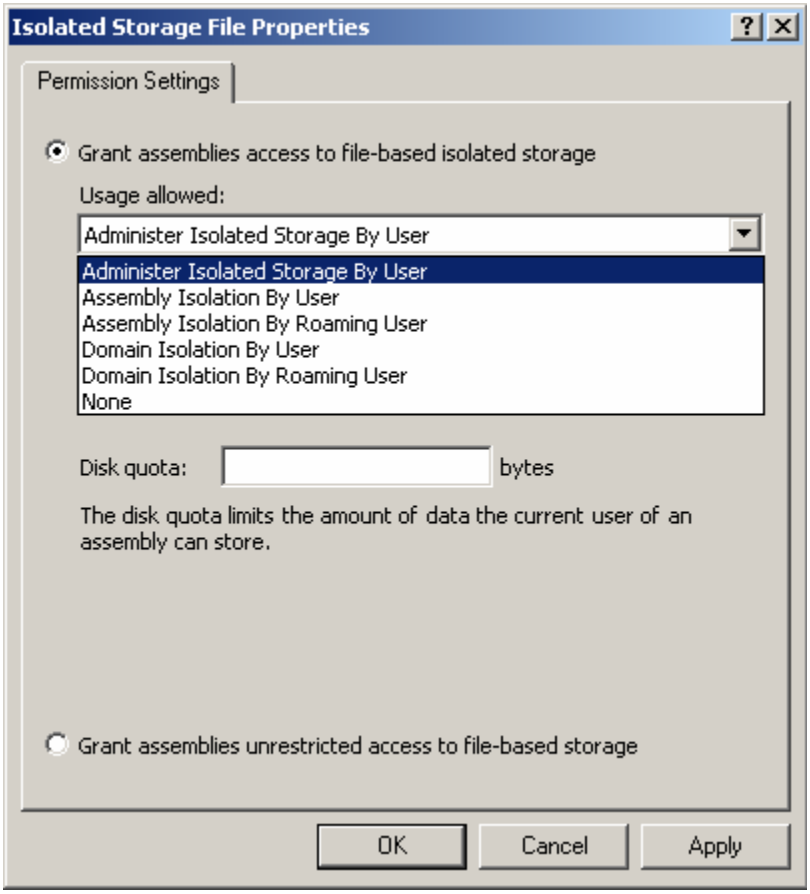
3.4.9.2 APPNET0003: Isolated Storage Permission

Description: The *Isolated Storage* permission is used to allow applications to store temporary data to a local user data store.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *IsolatedStorageFilePermission*

MSCORCFG.MSC: If the *Grant assemblies access to file-based isolated storage* is selected, view the selected *Usage allowed*.



Validate:

1. If the *Isolated Storage* permission of *Grant assemblies unrestricted access to file-based storage (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.
2. If the *Isolated Storage* permission *Administer Isolated Storage by User* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.
3. If the *Isolated Storage* permissions *Assembly Isolation by User* or *Assembly Isolation by Roaming User* is assigned to a Non-default Code Group whose membership criteria has not been evaluated and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0003	Isolated Storage Permission	
Reference:	.NET Framework Security Guide pg. 16	IA Control: DCSL-1

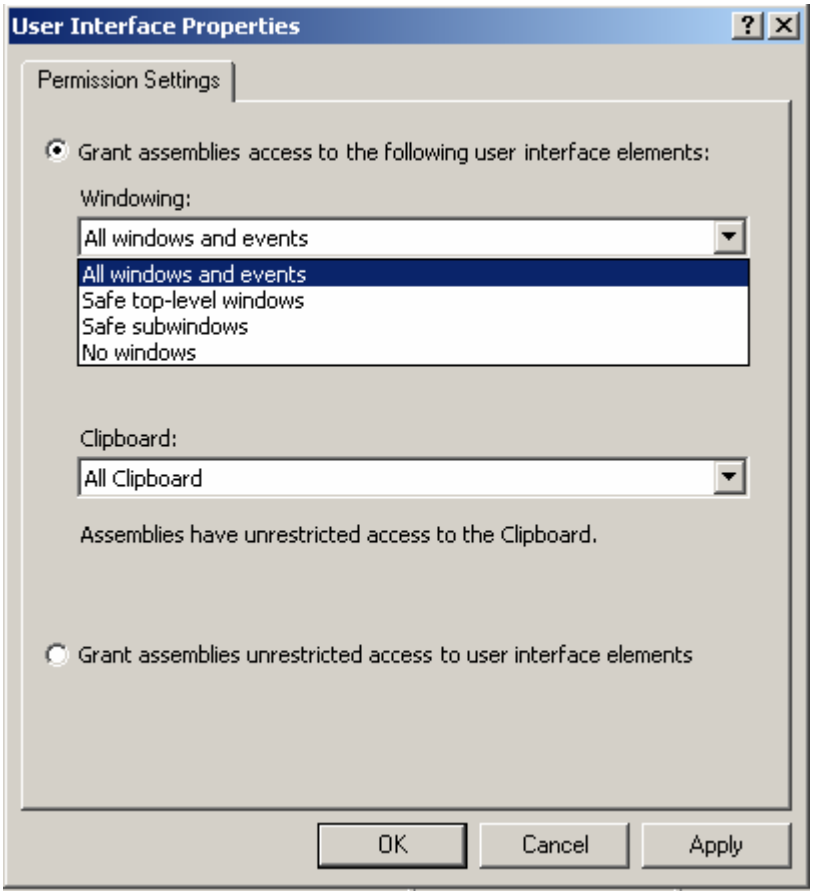
3.4.9.3 APPNET0004: User Interface Permission (Windowing)

Description: The User Interface Permission for windowing controls access to user interface windows.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *UIPermission*

MSCORCFG.MSC: If the *Grant assemblies access to the following user interface elements* is selected, view the selected *Windowing* restrictions selected.



Validate:

1. If the *User Interface* permission of *Grant assemblies unrestricted access to user interface elements (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..
2. If the *User Interface* permission *All Windows Events* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its' membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.
3. If the *User Interface* permissions *Safe Top Level Windows* is assigned to a Non-default Code Group whose membership criteria has not been evaluated and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0004	User Interface Permission (Windowing)	
Reference:	.NET Framework Security Guide pg 20	IA Control: DCSL-1

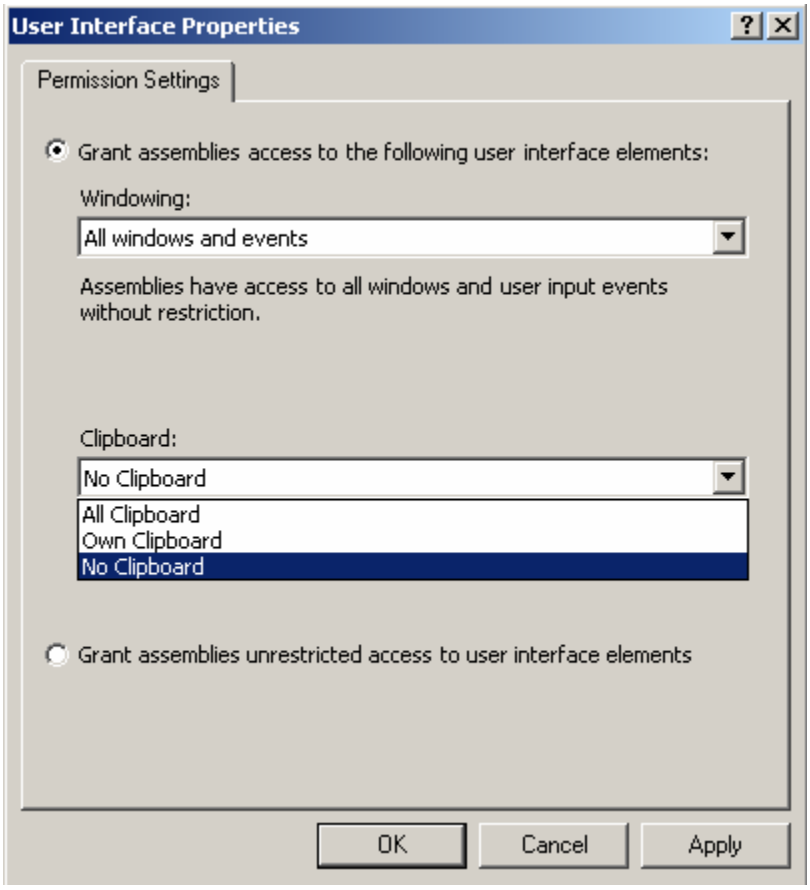
3.4.9.4 APPNET0005: User Interface Permission (Clipboard)

Description: The User Interface Permission for clipboard controls application access to clipboards used by the user or other applications.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *UIPermission*

MSCORCFG.MSC: If the *Grant assemblies access to the following user interface elements* is selected, view the selected *Clipboard* restrictions selected.



Validate:

1. If any *User Interface* permission other than *No Clipboard* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its' membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0005	User Interface Permission (Clipboard)	
Reference:	.NET Framework Security Guide pg 20	IA Control: DCSL-1

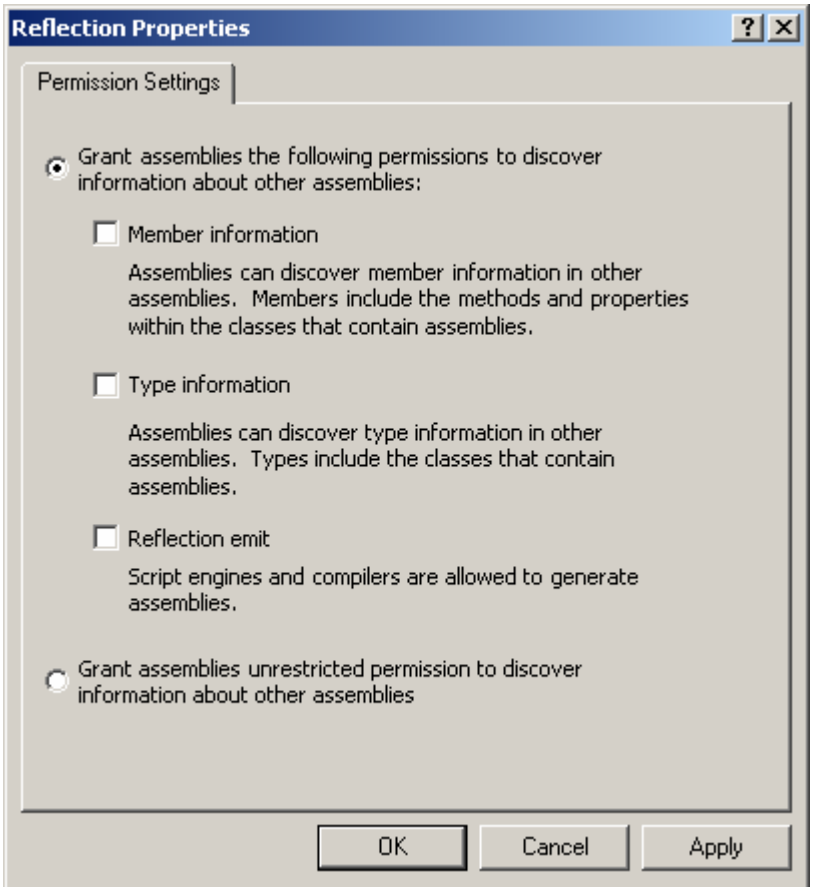
3.4.9.5 APPNET0006: Reflection Permission

Description: The Reflection permission controls an application's discovery of other system resources and applications.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *ReflectionPermission*

MSCORCFG.MSC: If the *Grant assemblies the following permissions to discover information about other assemblies* are selected, view the related selected checkboxes.



Validate:

1. If the *Reflection* permission of *Grant assemblies unrestricted permission to discover information about other assemblies (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..
2. If the *Reflection* permission *Member* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its' membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.
3. If the *Reflection* permissions *Type* is assigned to a Non-default Code Group whose membership criteria has not been reviewed and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0006	Reflection Permission	
Reference:	.NET Framework Security Guide pg 21	IA Control: DCSL-1

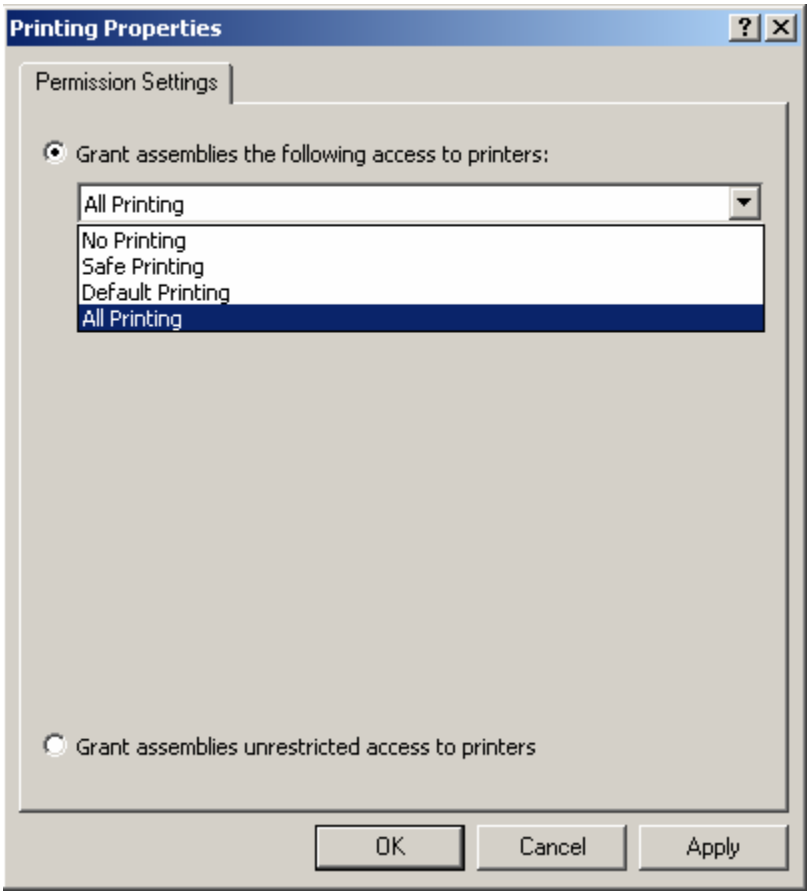
3.4.9.6 APPNET0007: Printing Permission

Description: The Printing permission controls application access to system printing resources.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *PrintingPermission*

MSCORCFG.MSC: If the *Grant assemblies the following access to printers* is selected, view the access selected.



Validate:

1. If the *Printing* permission of *Grant assemblies unrestricted access to printers (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..
2. If the *Printing* permission *All Printing* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its' membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0007	Printing Permission	
Reference:	.NET Framework Security Guide pg 22	IA Control: DCSL-1

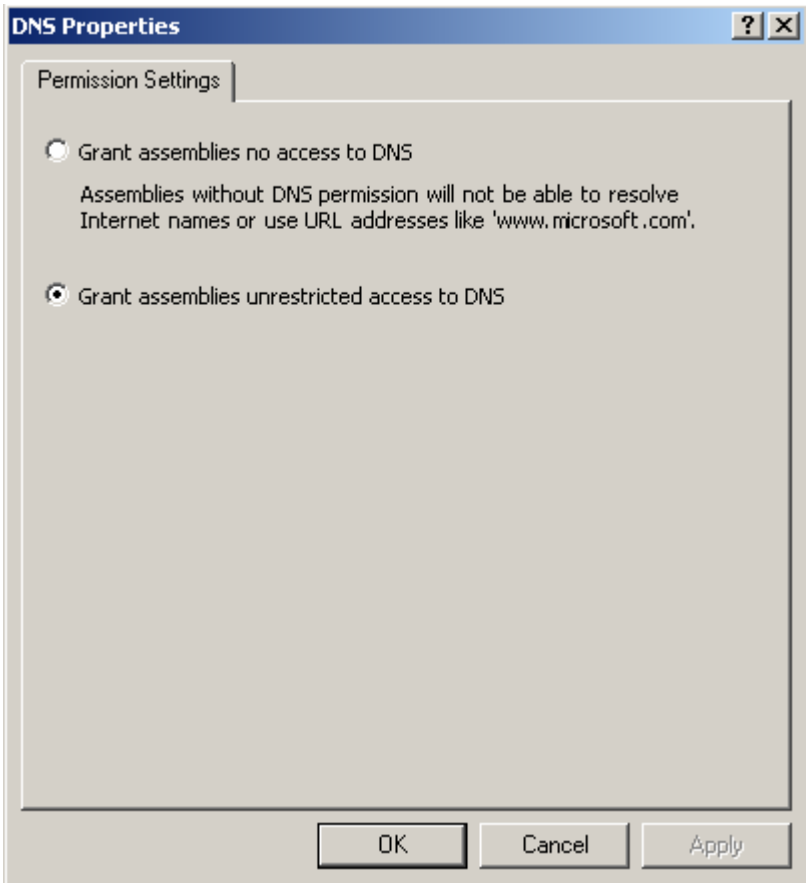
3.4.9.7 APPNET0008: DNS Permission

Description: The DNS permission controls application access to DNS resources available to the host system.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *DNSPermission*

MSCORCFG.MSC:



Validate:

1. If the *DNS* permission of *Grant assemblies unrestricted access to DNS (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0008	DNS Permission	
Reference:	.NET Framework Security Guide pg 23	IA Control: ECLP-1

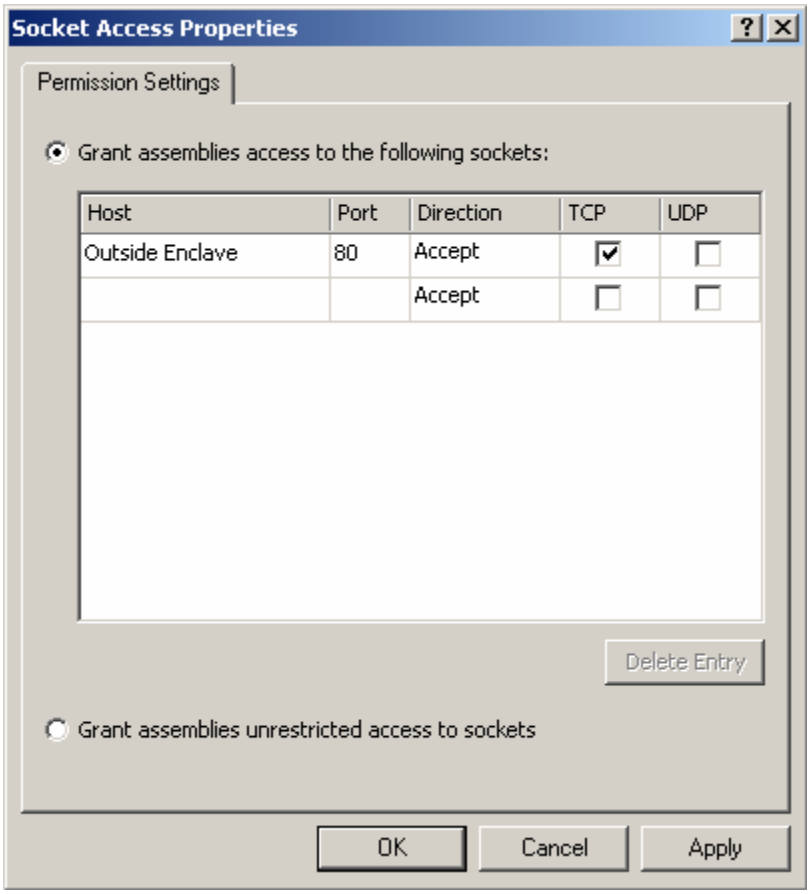
3.4.9.8 APPNET0009: Socket Access Permission

Description: The Socket Access permission controls application access to network ports defined on the host system.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *SocketPermission*

MSCORCFG.MSC: If the *Grant assemblies access to following sockets* is selected, view the listed sockets.



Validate:

1. If the *Socket Access* permission of *Grant assemblies unrestricted access to sockets (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..
2. Ask the System Administrator if any *Socket Access* permissions are granted to Non-default Code groups that do not provide networking services. If these permissions exist then this is a finding.
3. Ask the System Administrator if any *Socket Access* permissions are granted to Non-default Code groups to hosts outside the enclave. If these permissions exist then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0009	Socket Access Permission	
Reference:	.NET Framework Security Guide pg 23	IA Control: DCSL-1

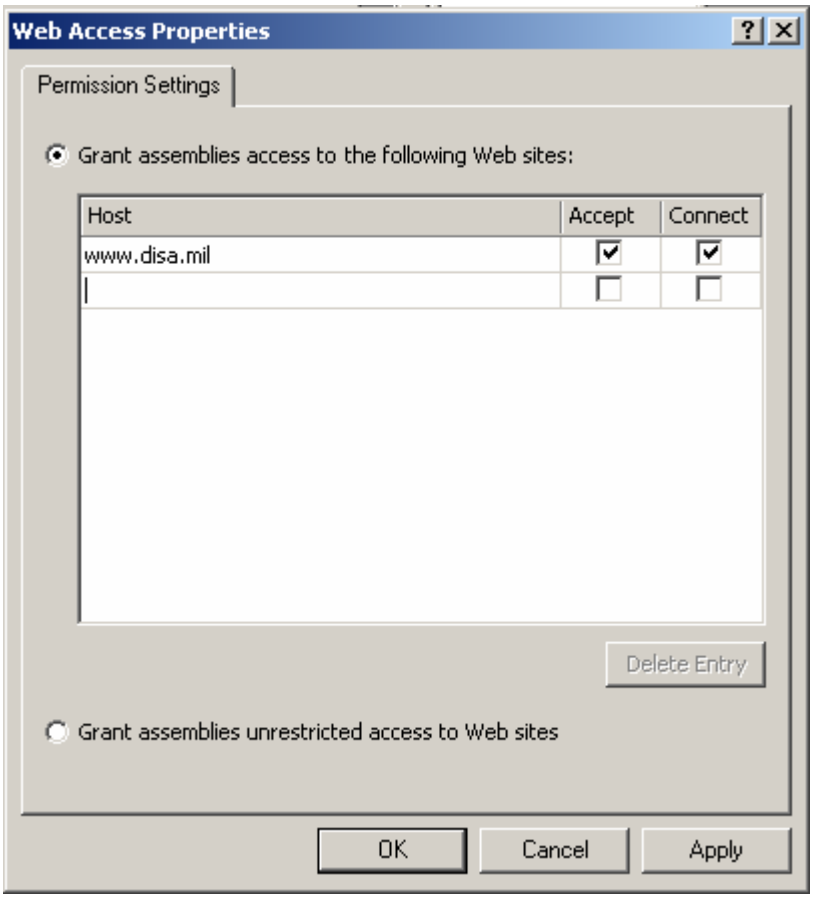
3.4.9.9 APPNET0010: Web Access Permission

Description: The Web Access permission controls application access to HTTP requests to designated URLs or the configuration of HTTP settings.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *WebPermission*

MSCORCFG.MSC: If the *Grant assemblies access to the following Web sites* is selected, note the listed web sites.



Validate:

1. If the *Web Access* permission of *Grant assemblies unrestricted access to Web Sites (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..
2. If specific URL(s) (*Web Access* permissions) are assigned to a Non-default Code Group whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET10	Web Access Permission	

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
Reference:	.NET Framework Security Guide pg 24	IA Control: DCSL-1

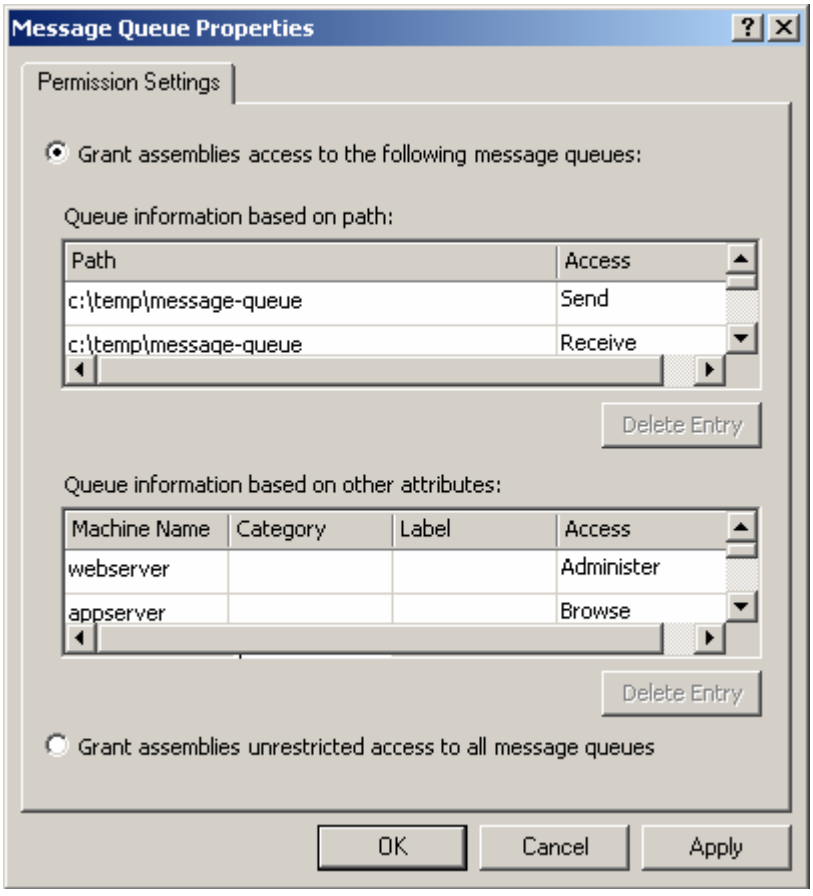
3.4.9.10 APPNET0011: Message Queue Permission

Description: The Message Queue permission controls application access to communications across the network.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *MessageQueuePermission*

MSCORCFG.MSC: If the *Grant assemblies access to the following message queues* is selected, note the queues listed and the access granted to them.



Validate:

1. If the *Message Queue* permission of *Grant assemblies unrestricted access to all message queues (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.
2. If the *Message Queue* permission *Administer* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership

condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..

3. If the *Message Queue* permission *Browse* is assigned to a Non-default Code Group whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0011	Message Queue Permission	
Reference:	.NET Framework Security Guide pg 25	IA Control: ECLP-1

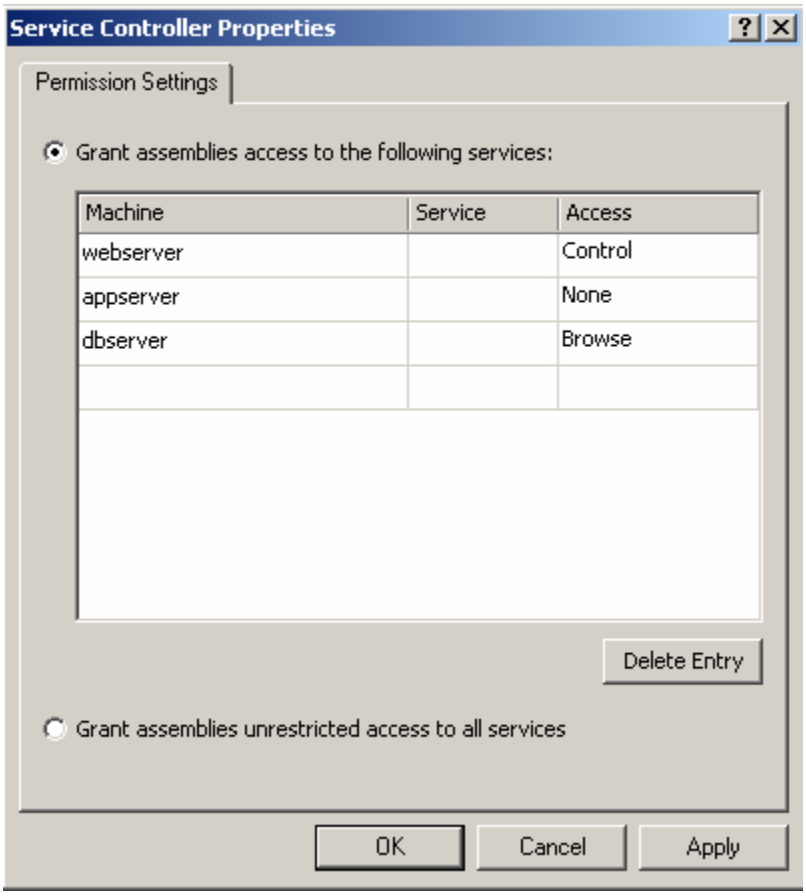
3.4.9.11 APPNET0012: Service Controller Permission

Description: The Service Controller permission controls application access to the control of Windows services.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *ServiceControllerPermission*

MSCORCFG.MSC: If the *Grant assemblies access to the following services* is selected, note the services listed and the access granted to them.



Validate:

1. If the *Service Controller* permission of *Grant assemblies unrestricted access to all services (unrestricted=”true”)* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..
2. If the any *Service Controller* permissions are assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as its’ membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0012	Service Controller Permission	
Reference:	.NET Framework Security Guide pg 25	IA Control: DCSL-1

3.4.9.12 APPNET0013: Database Permission

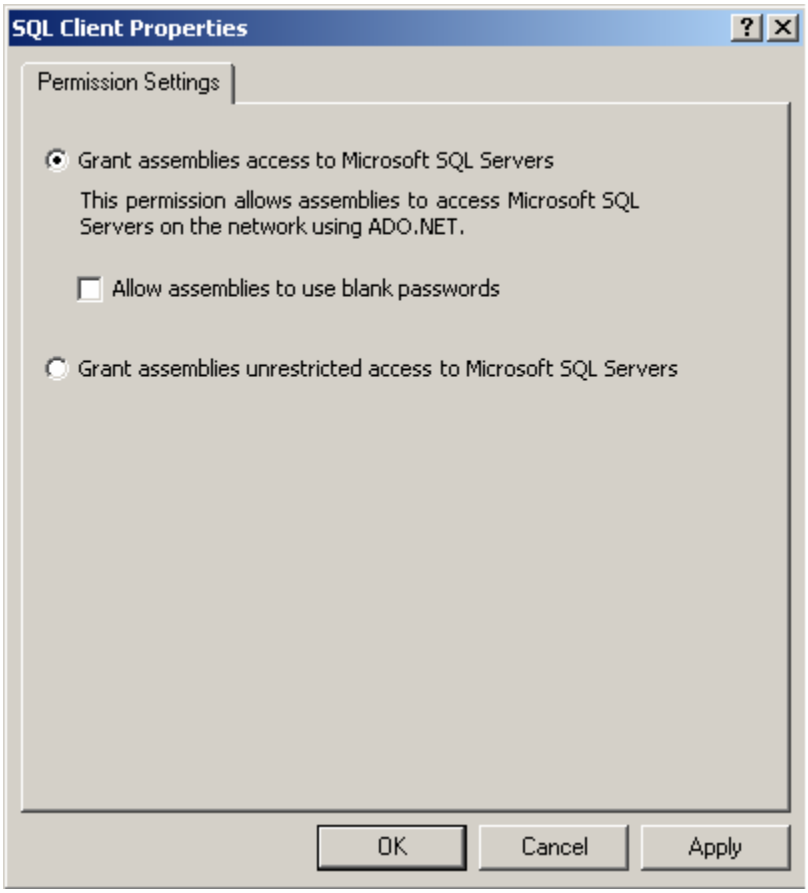
Description: The Database permissions control application access to databases defined on the host system.

Applies to: Version 1.0; Version 1.1; Version 2.0

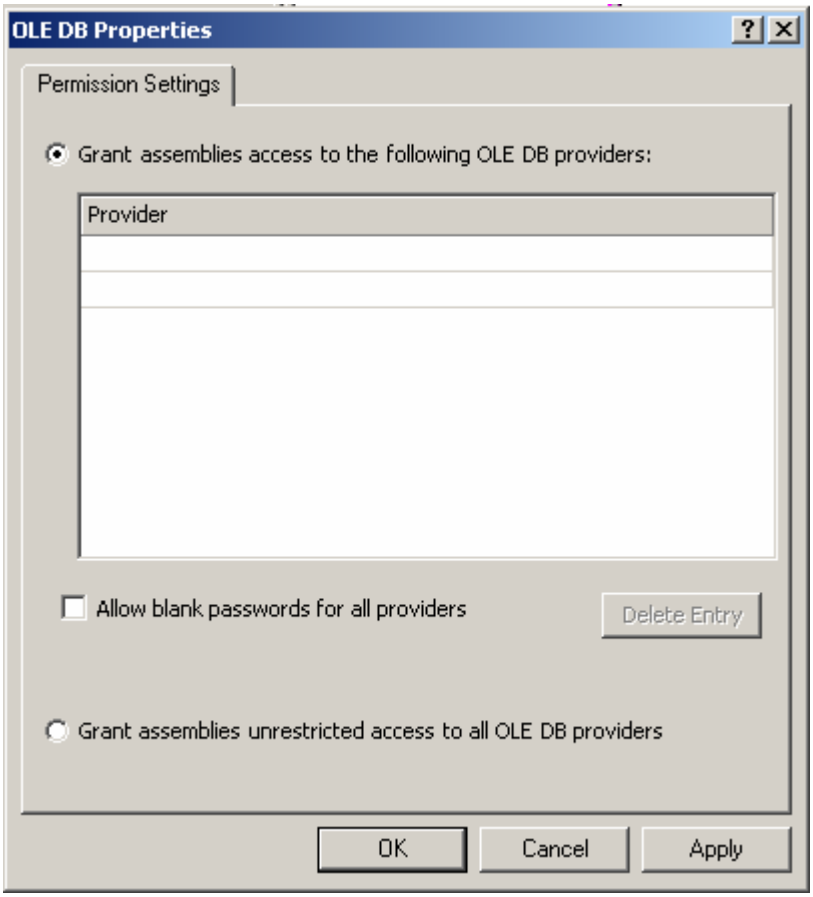
CASPOL.EXE: Permission Names:
SQLClientPermission,
OLEDBPermission

MSCORCFG.MSC: If the *Grant assemblies access to following providers* is selected, note the selected *Providers*.

SQLClient Permisssion



OLEDB Permission

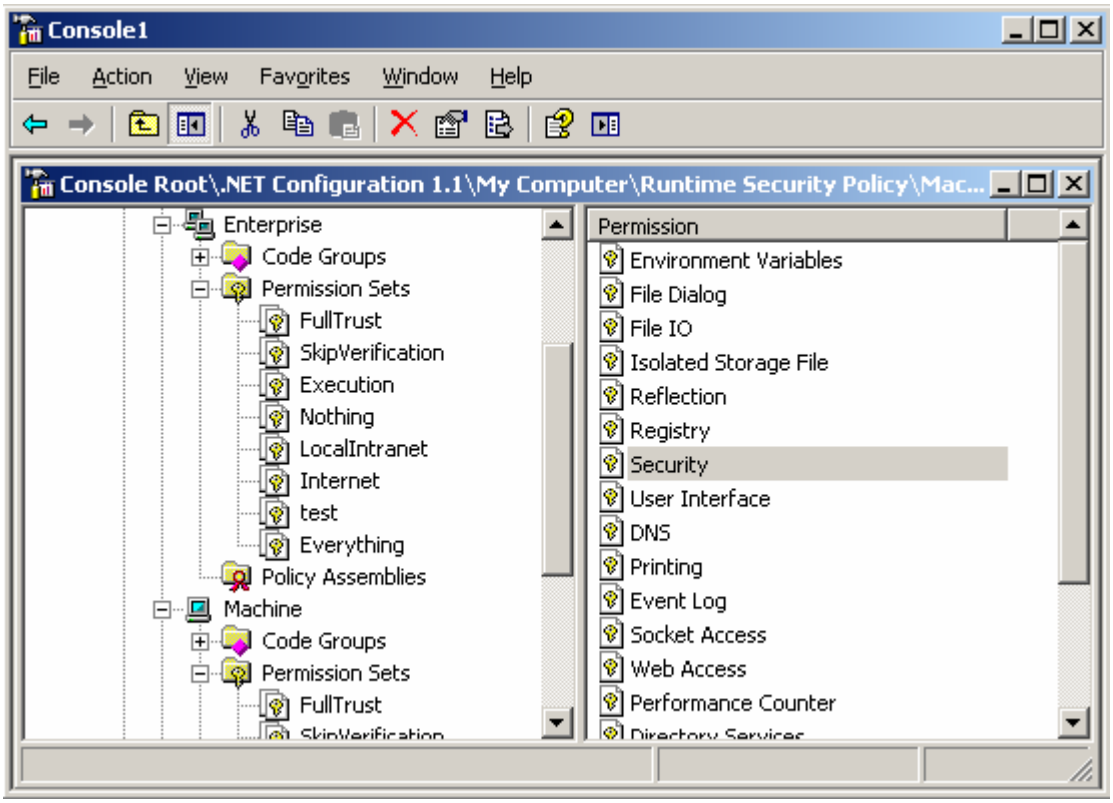


Validate:

1. If the *SQLClientPermission or OLEDBPermission*permission (*Grant assemblies unrestricted access to all providers (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0013	Database Permission	
Reference:	.NET Framework Security Guide pg 33	IA Control: DCSL-1

Checks APPNET014 – APPNET025 refer to the Security permission contained within a permission set assigned to a non-default code group. Review all permission sets that include the security permission.



3.4.9.13 APPNET0014: Security Permission (Extend Infrastructure)

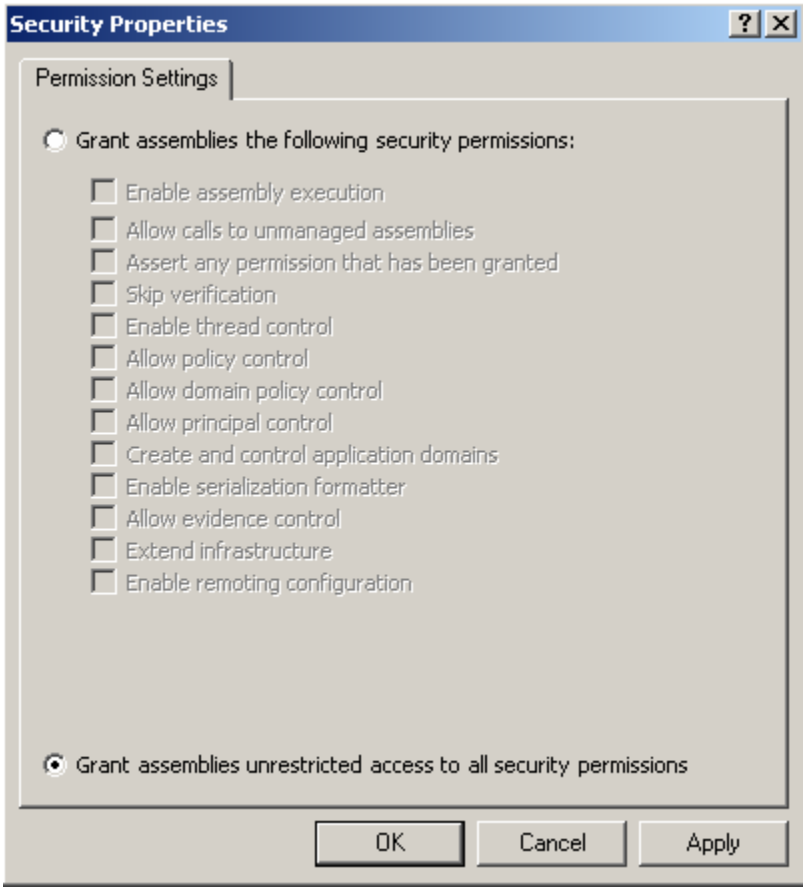
Description: The Security permission Extend Infrastructure controls application access to message processing.

Applies to: Version 1.0; Version 1.1; Version 2.0

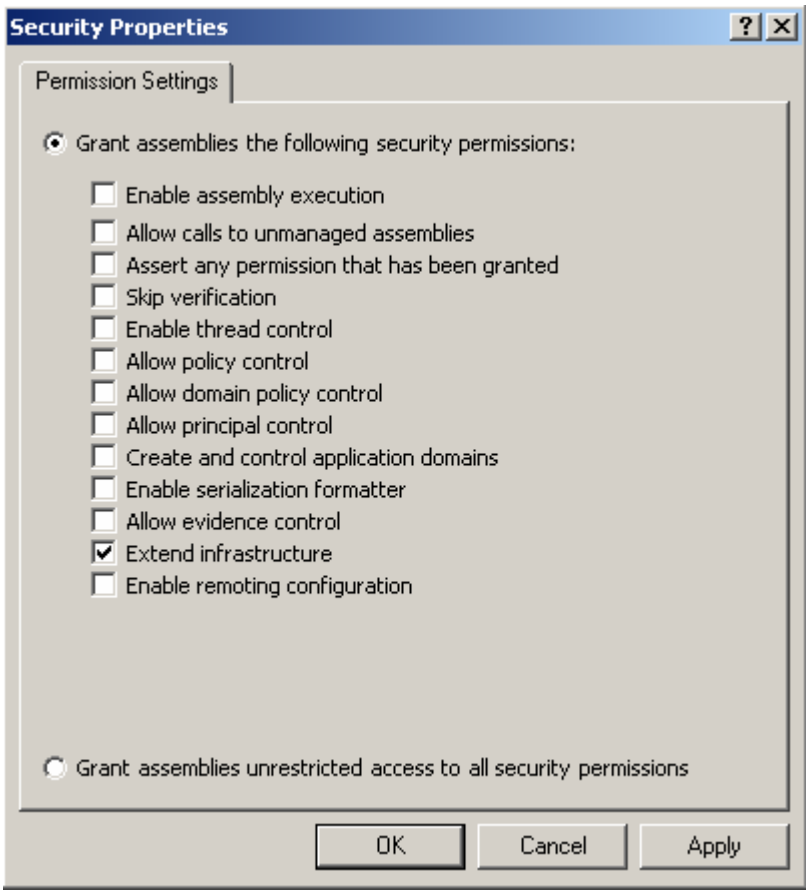
CASPOL.EXE: Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Grant assemblies unrestricted access to all security permission (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



2. If the *Security* permission of *Extend Infrastructure* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0014	Security Permission (Extend Infrastructure)	
Reference:	.NET Framework Security Guide 35	IA Control: DCSL-1

3.4.9.14 APPNET0015: Security Permission (Enable Remoting Configuration)

Description: The Security permission Enable Remoting Configuration defines the communication channels available to an application.

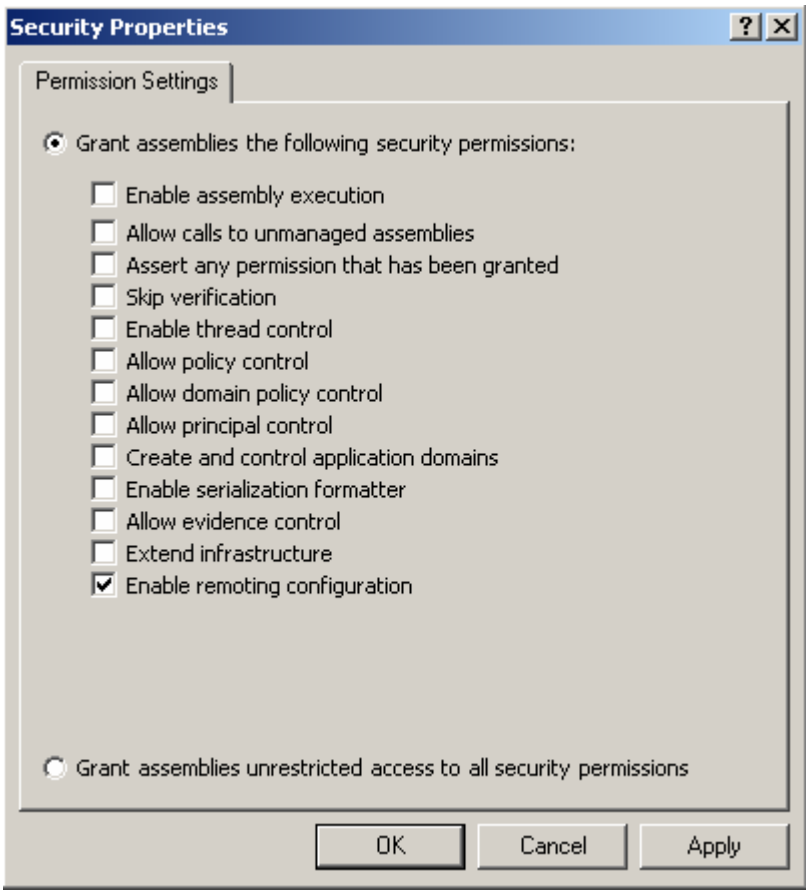
Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE:
Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Enable remoting configuration (Flags="RemotingConfiguration")* is assigned to a Non-default Code Group that

does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0015	Security Permission (Enable Remoting configuration)	
Reference:	.NET Framework Security Guide pg 35	IA Control: DCSL-1

3.4.9.15 APPNET0016: Security Permission (Enable Serialization Formatter)

Description: The Security permission Enable Serialization Formatter controls access to serialized data. Serialized data is data formatted into a series of bits for storing or transmitting.

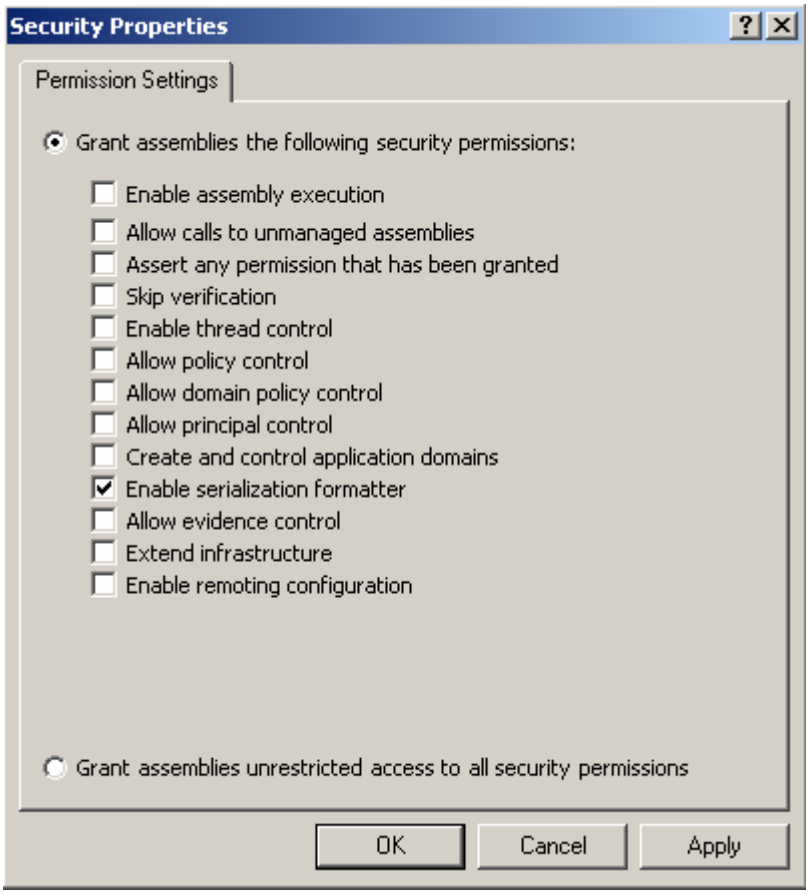
Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE:
Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Enable Serialization Formatter (Flags=" SerializationFormatter")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and

whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0016	Security Permission (Enable Serialization Formatter)	
Reference:	.NET Framework Security Guide pg 36	IA Control: DCSL-1

3.4.9.16 APPNET0017: Security Permission (Enable Thread Control)

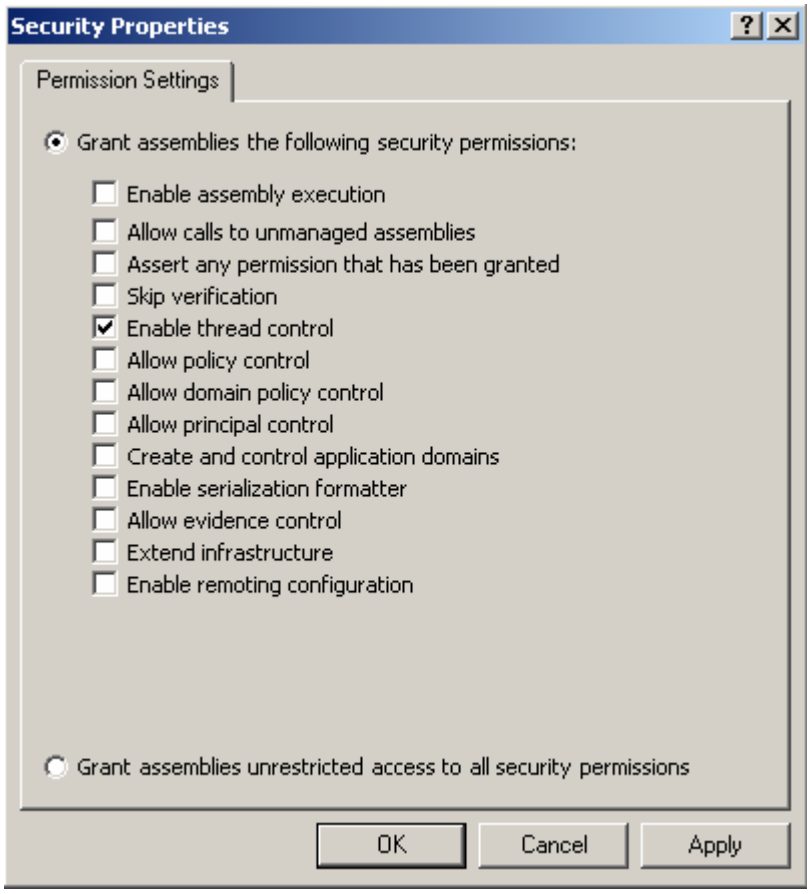
Description: The Security permission Enable Thread Control is used to control application access to abort, suspend, or resume its threads.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE:
Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Enable thread control (Flags="ControlThread")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0017	Security Permission (Enable Thread Control)	
Reference:	.NET Framework Security Guide pg 36	IA Control: DCSL-1

3.4.9.17 APPNET0018: Security Permission (Allow Principal Control)

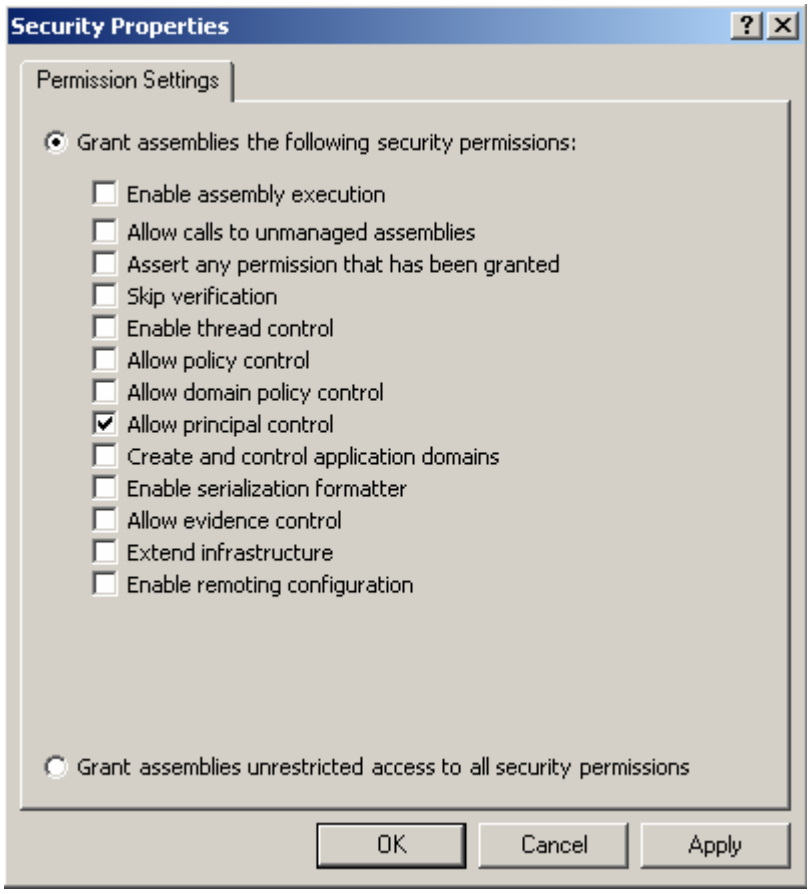
Description: The Security permission Allow Principal control controls application access to Windows user information.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Allow principal control (Flags="ControlPrincipal")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0018	Security Permission (Allow Principal Control)	
Reference:	.NET Framework Security Guide pg 37	IA Control: DCSL-1

3.4.9.18 APPNET0019: Security Permission (Enable Assembly Execution)

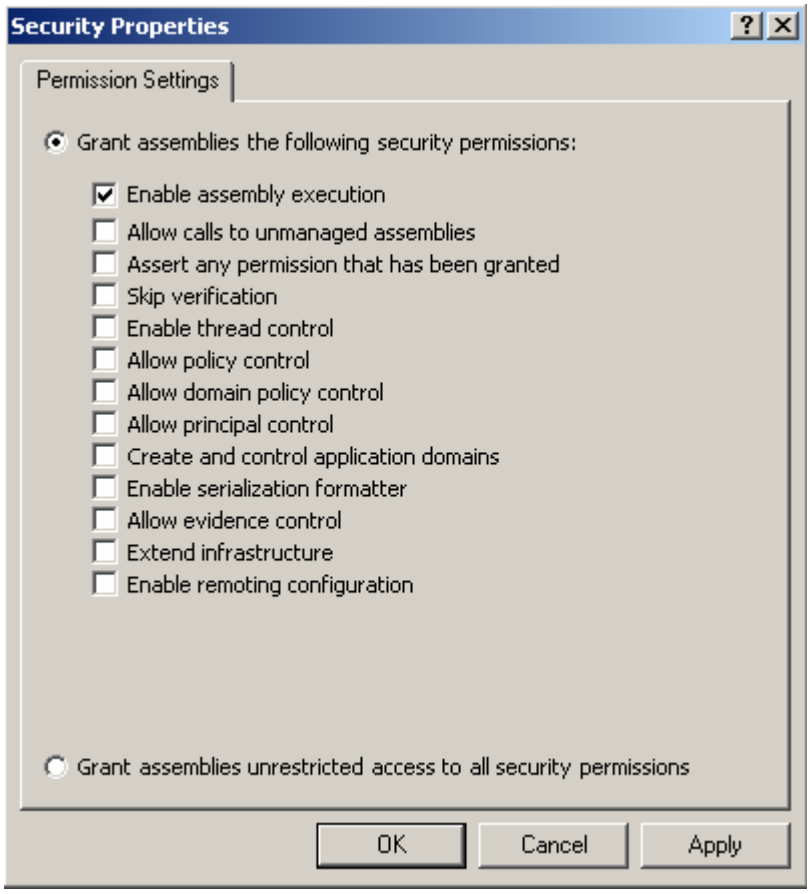
Description: The Security permission Enable Assembly Execution allows applications to execute.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Enable assembly execution* (*Flags="Execution")* is in a permission set that is assigned to a Non-default Code Group with a *Zone* membership condition then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0019	Security Permission (Enable Assembly Execution)	
Reference:	.NET Framework Security Guide pg 38	IA Control: ECLP-1

3.4.9.19 APPNET0020: Security Permission (Skip Verification)

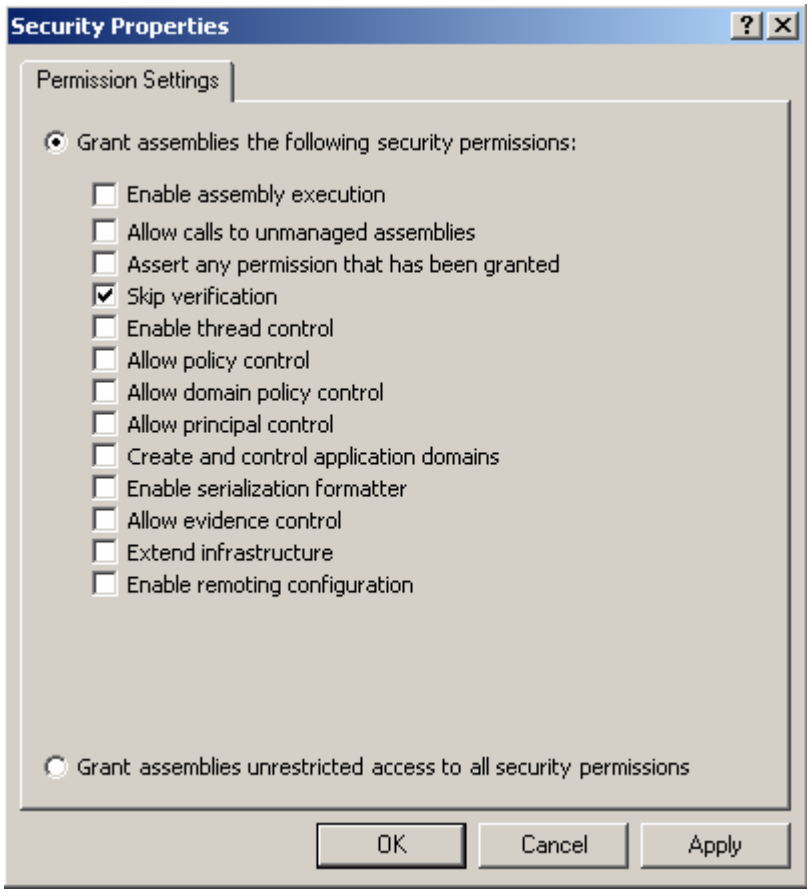
Description: The Security permission Skip Verification controls the execution of code that is verified as being type safe.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Skip verification (Flags="SkipVerification")* is assigned to any non-default or default Code Group then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0020	Security Permission (Skip Verification)	
Reference:	.NET Framework Security Guide pg 39	IA Control: ECLP-1

3.4.9.20 APPNET0021: Security Permission (Allow Calls to Unmanaged Assemblies)

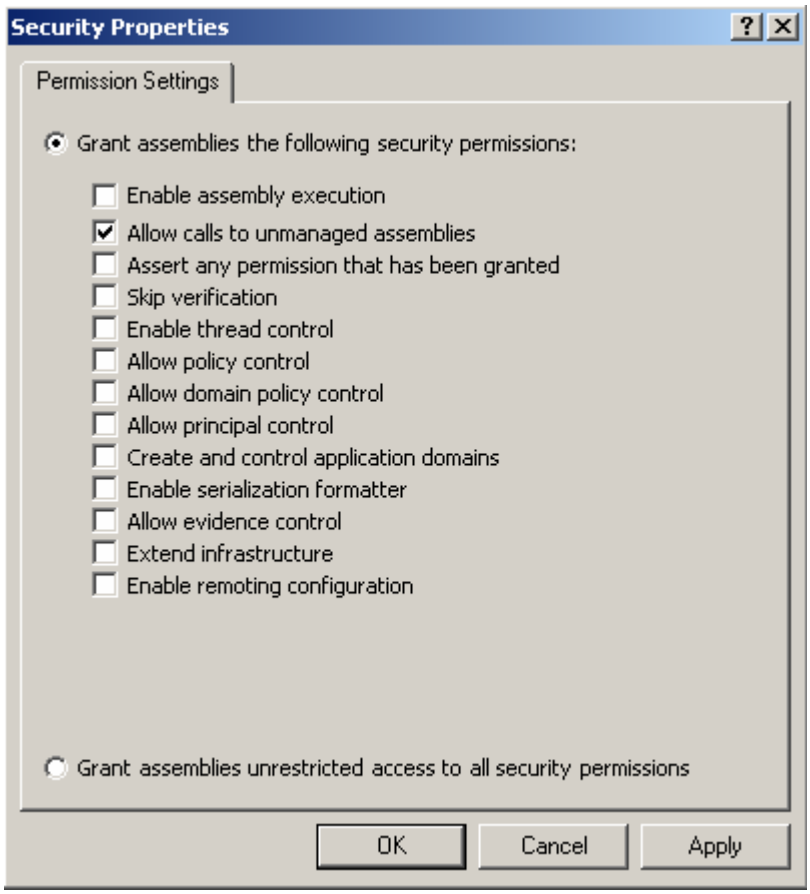
Description: The Security permission Allow Calls to Unmanaged Assemblies controls application access to applications not managed by the .Net Framework.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Allow calls to unmanaged assemblies (Flags="UnmanagedCode")* is assigned to a non-default Code Group then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0021	Security Permission (Allow Calls to Unmanaged Assemblies)	
Reference:	.NET Framework Security Guide pg 40	IA Control: DCSL-1

3.4.9.21 APPNET0022: Security Permission (Allow Policy Control)

Description: The Security permission Allow Policy Control controls application access to it's the current security policy configuration.

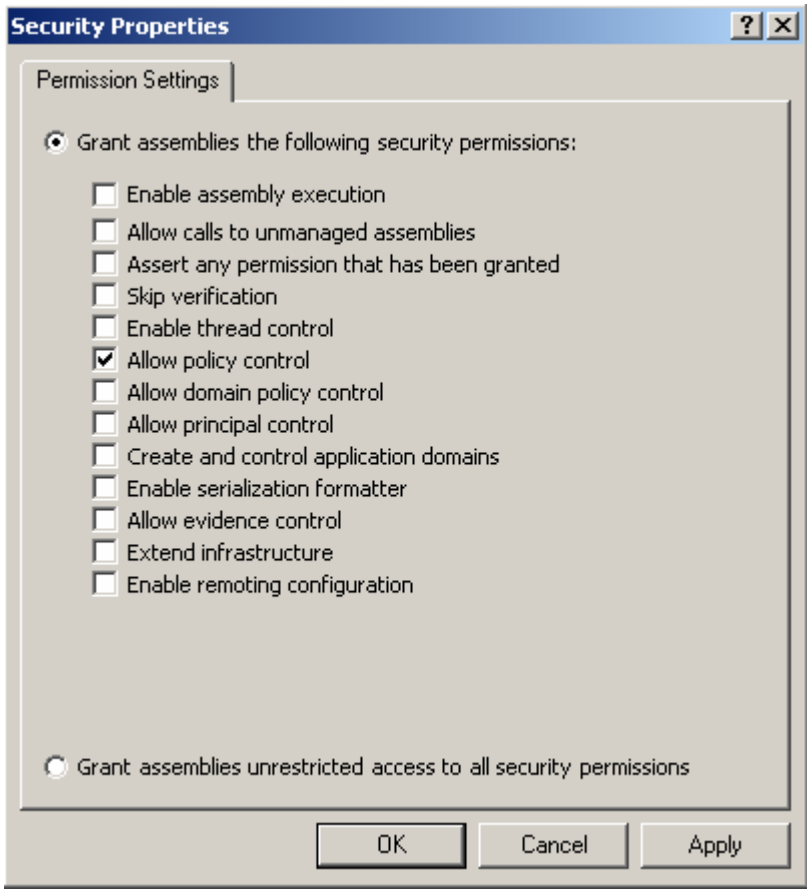
Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE:
Permission Name: *SecurityPermission*

MSCORCFG.MSC: If the *Grant assemblies the following security permissions* is selected, note the selected permissions.

Validate:

1. If the *Security* permission of *Allow Policy Control (Flags="ControlPolicy")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0022	Security Permission (Allow Policy Control)	
Reference:	.NET Framework Security Guide pg 40	IA Control: DCSL-1

3.4.9.22 APPNET0023: Security Permission (Allow Domain Policy Control)

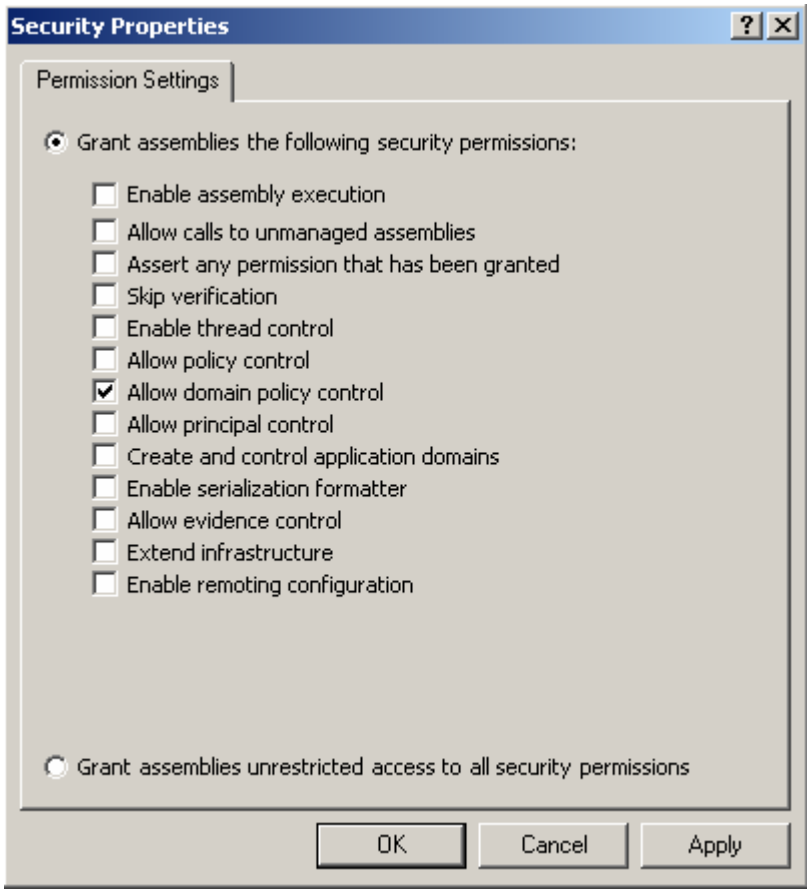
Description: The Security permission Allow Domain Policy controls defines application access to its own application domain security policy.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE:
Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Allow domain policy control (Flags="ControlDomainPolicy")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0023	Security Permission (Allow Domain Policy Control)	
Reference:	.NET Framework Security Guide pg 41	IA Control: DCSL-1

3.4.9.23 APPNET0024: Security Permission (Allow Evidence Control)

Description: The Security permission Allow Evidence Control is used to control an application's access to supply or modify evidence used to determine access to system resources.

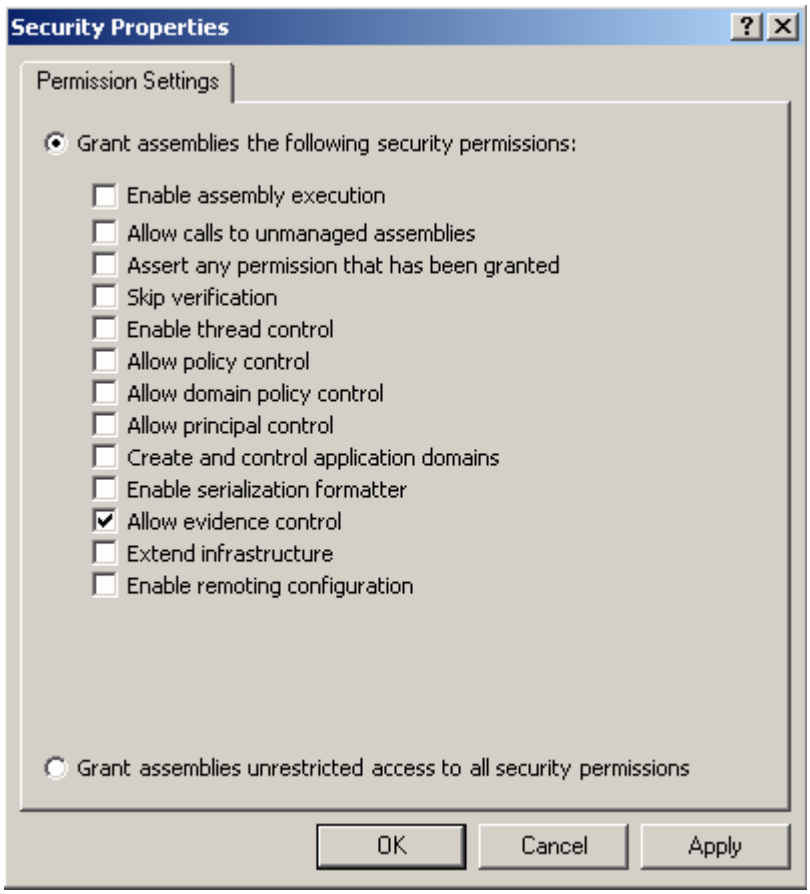
Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE:

Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Allow evidence control (Flags="ControlEvidence")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0024	Security Permission (Allow Evidence Control)	
Reference:	.NET Framework Security Guide pg 41	IA Control: DCSL-1

3.4.9.24 APPNET0025: Security Permission (Assert any Permission that Has Been Granted)

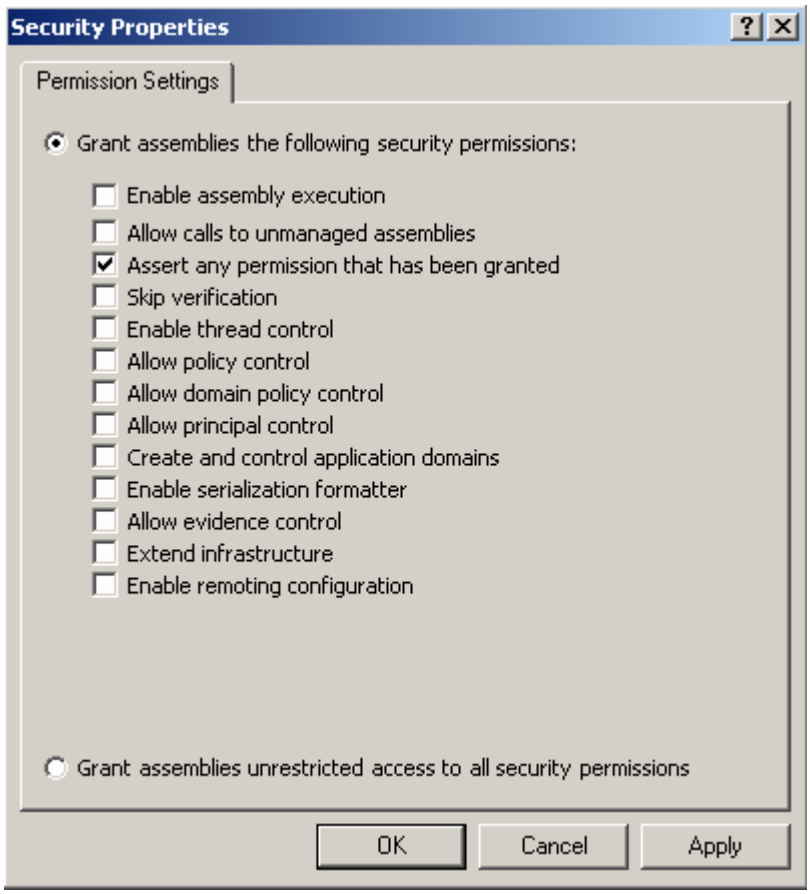
Description: The Security permission Assert any Permission that Has Been Granted controls application access to permissions assigned to any code in the assembly that called it.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *SecurityPermission*

Validate:

1. If the *Security* permission of *Assert (Flags="Assertion")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0025	Security Permission (Assert any Permission that Has Been Granted)	
Reference:	.NET Framework Security Guide pg 42	IA Control: DCSL-1

3.4.9.25 APPNET0026: Performance Counter Permission

Description: The Performance Counter permission controls application access to system performance monitoring resources.

Applies to: Version 1.0; Version 1.1; Version 2.0

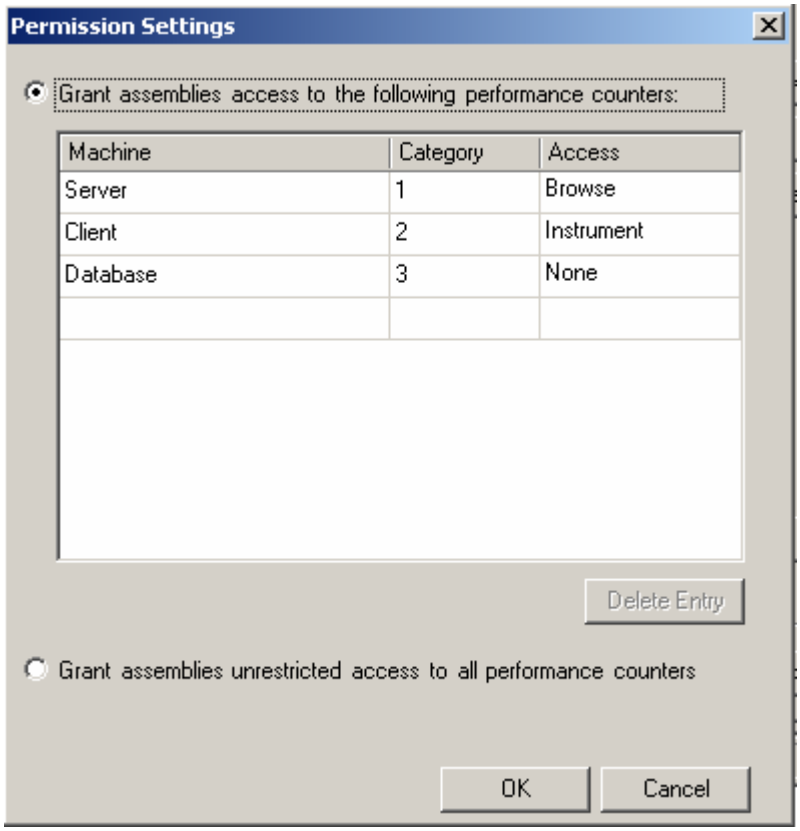
CASPOL.EXE: Permission Name: *PerformanceCounterPermission*

MSCORCFG.MSC: If the *Grant assembly's access to the following performance counters* is selected, note the listed counters.

Validate:

1. If the *Performance Counter* permission of *Grant assemblies unrestricted access to performance counters (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding..

2. If the *Performance Counter* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0026	Performance Counter Permission	
Reference:	.NET Framework Security Guide pg 44	IA Control: ECLP-1

3.4.9.26 APPNET0027: Environment Variables Permission

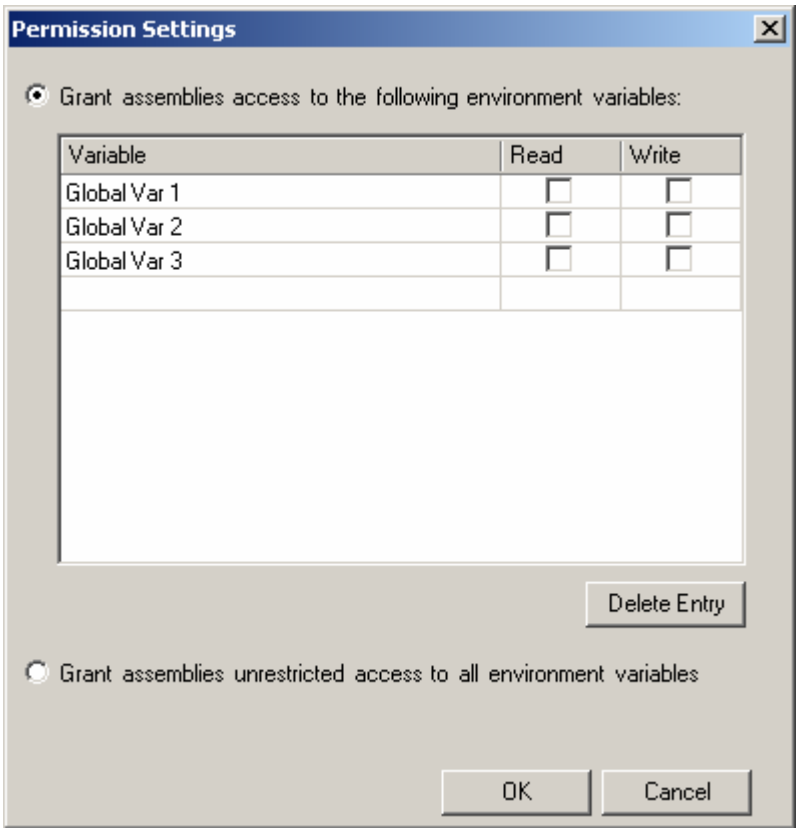
Description: The Environment Variables permission controls application access to system environment variables and to other system resource names.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *EnvironmentPermission*

Validate:

1. If the *Environment Variables* permission of *Grant assemblies access to all environment variables (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0027	Environment Variables Permission	
Reference:	.NET Framework Security Guide pg 48	IA Control: ECLP-1

3.4.9.27 APPNET0028: Event Log Permission

Description: The Event Log permission controls application access to event log resources defined on the system.

Applies to: Version 1.0; Version 1.1; Version 2.0

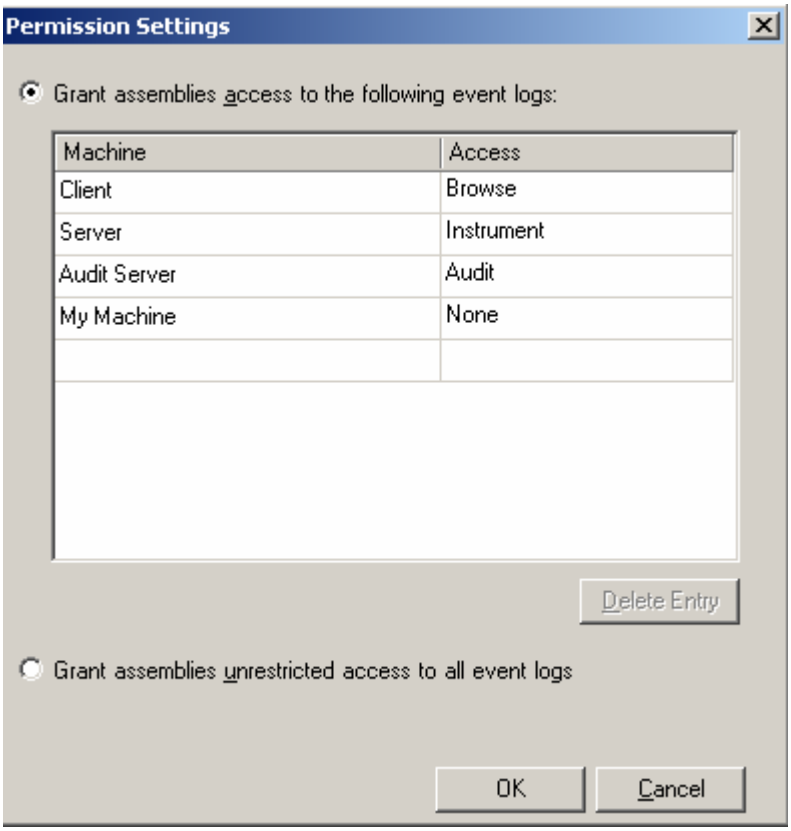
CASPOL.EXE: Permission Name: *EventLogPermission*

MSCORCFG.MSC: If the *Grant assemblies access to following event logs* is selected, note the selected machines and access type.

Validate:

1. If the *Event Log* permission of *Grant assemblies unrestricted access to all event logs (unrestricted="true")* is assigned to a Non-default Code Group that does not use a

Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0028	Event Log Permission	
Reference:	.NET Framework Security Guide pg 49	IA Control: ECLP-1

3.4.9.28 APPNET0029: Registry Permission

Description: The Registry permission controls application access to the Windows registry.

Applies to: Version 1.0; Version 1.1; Version 2.0

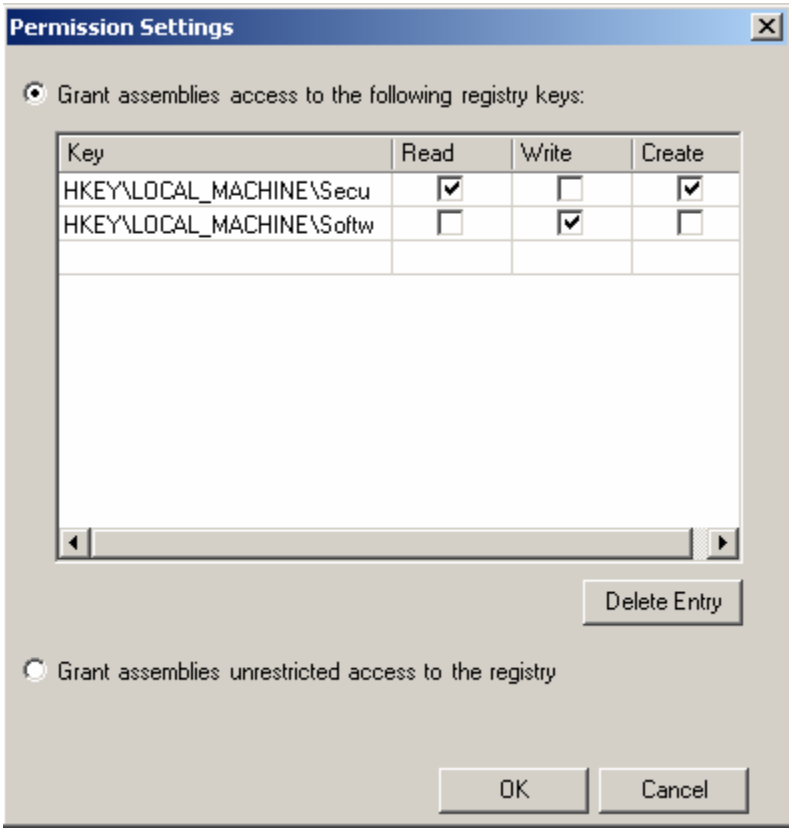
CASPOL.EXE: Permission Name: *RegistryPermission*

MSCORCFG.MSC: If the *Grant assemblies access to the following registry keys* is selected, note the selected registry keys and the access assigned.

Validate:

1. If a Permission Set with the *Registry* permission of *Grant Assemblies unrestricted access to the registry (unrestricted="true")* is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.

2. If a Permission Set granting limited *Registry* permissions is assigned to a Non-default Code Group that does not use a Strong Name, Publisher, or Hash as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0029	Registry Permission	
Reference:	.NET Framework Security Guide pg 49	IA Control: DCSL-1

3.4.9.29 APPNET0030: Directory Services Permission

Description: The Directory Services permission controls application access to the system Directory Service resources.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Permission Name: *DirectoryServicesPermission*

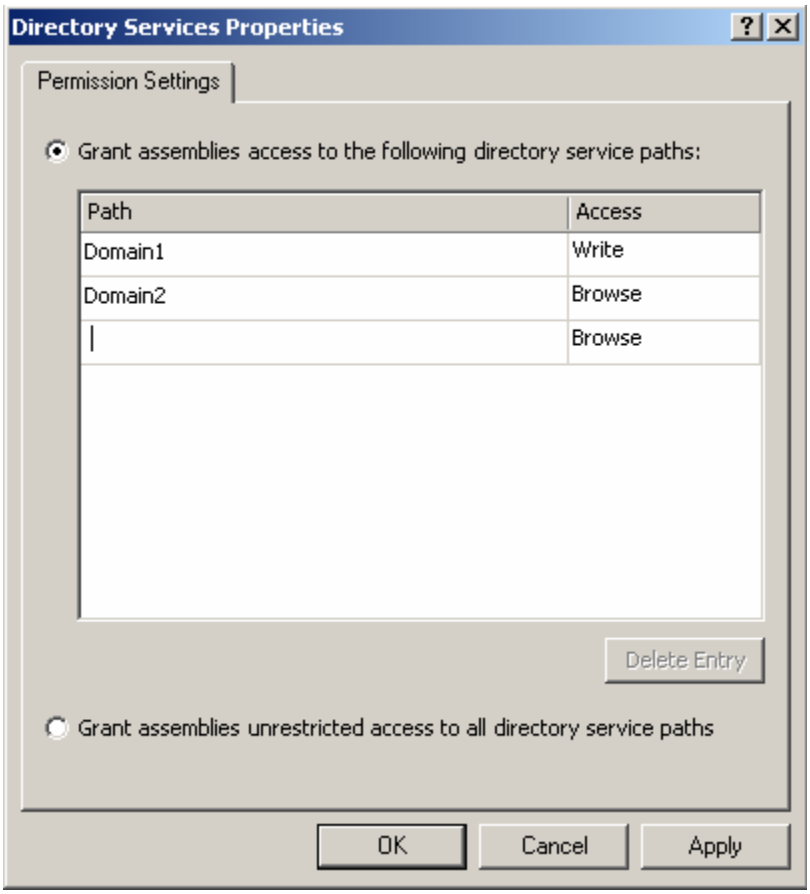
MSCORCFG.MSC: If the *Grant assemblies access to the following directory service paths* is selected, note the listed paths and the access assigned.

Validate:

1. If the *Directory Services* permission of *Grant assemblies unrestricted access to all directory service paths (unrestricted="true")* is assigned to a Code Group that does

not use a Strong Name as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.

2. If the *Directory Services* permission of *Write* () or Browse() is assigned to a Code Groups that does not use a Strong Name as the membership condition and whose assignment criteria has not been reviewed and approved by the IAO then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0030	Directory Services Permission	
Reference:	.NET Framework Security Guide pg 50	IA Control: DCSL-1

3.4.9.30 APPNET0031: Strong Name Membership Condition

Description: The Strong Name Membership Condition establishes the requirement for all code defined in the group to be configured with a Strong Name. Strong Name verification should not be omitted in a production environment.

Applies to: Version 1.0; Version 1.1; Version 2.0

SN.EXE: Review the sn.exe listing for assemblies skipping strong name verification. A delay-signed assembly may be stored in a list in the registry entry HKLM\Software\Microsoft\StrongName\Verification. Execute sn.exe –Vl option.

Sample Non Finding Example:
sn.exe -Vl

Microsoft (R) .NET Framework Strong Name Utility Version 1.1.4322.573
Copyright (C) Microsoft Corporation 1998-2002. All rights reserved.

No verification entries registered

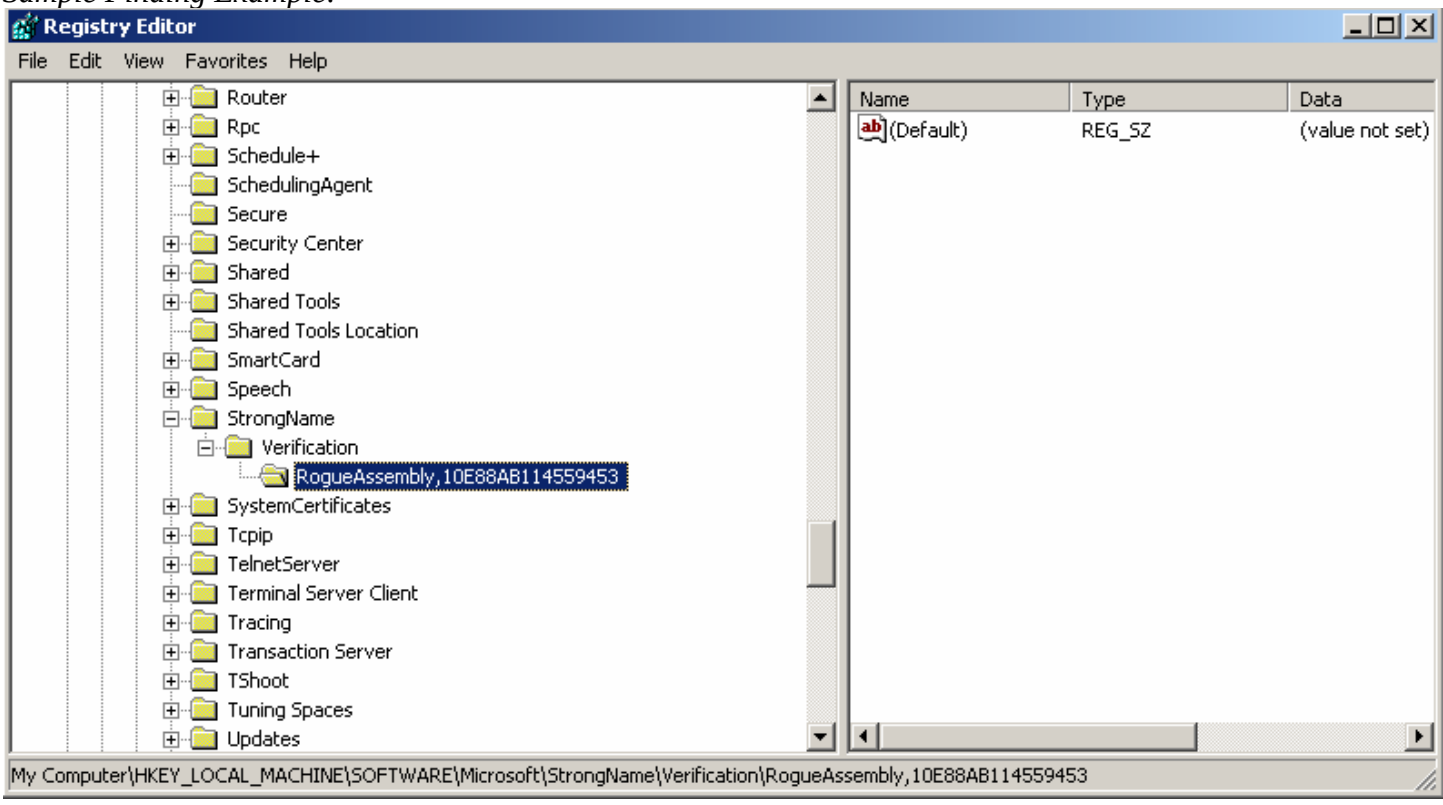
Sample Finding Example:
sn -Vl

Microsoft (R) .NET Framework Strong Name Utility Version 1.1.4322.573 Copyright (C) Microsoft Corporation 1998-2002. All rights reserved.

Assembly/Strong Name Users
==========
RogueAssembly,10E88AB114559453 All users

REGEDIT: Review the Windows Registry for assemblies omitting strong name verification in the registry entry HKLM\Software\Microsoft\StrongName\Verification. There should be no assemblies list under this registry key.

Sample Finding Example:



Validate:

1. If any assemblies are listed as omitting Strong Name verification in a production environment then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0031	Strong Name Membership Condition	
Reference:	.NET Framework Security Guide pg 57	IA Control: DCSL-1

3.4.9.31 APPNET0032: First Match Code Groups

Description: The First Match Code Group is used to control the depth to which a branch of the code group tree is traversed when assigning membership to assemblies.

Applies to: Version 1.0; Version 1.1; Version 2.0

Check for the CAS configuration file for First Match Code Groups.
Look for any entries in the configuration file similar to the following:
<CodeGroup class="FirstMatchCodeGroup"

Configuration File Location:
Enterprise Level: %WINDIR%\Microsoft.Net\Framework\vx.y.zzzz\config\EnterpriseSec.config
Machine Level: %WINDIR%\Microsoft.Net\Framework\vx.y.zzzz\config\Security.config
User Level: %APPDATA%\Microsoft\CLR Security Config\vx.y.zzzz.ww\Security.config
The vx.y.zzzz are version strings for the .Net Framework

v1.0.3705
v1.1.4322
v2.0.50727 (v2.0.50727.832 for the user level)

Validate any use of First Match Code Groups was intentional and that the implementation of the First Match Code Groups is correct. Ask the System Administratorto validate the security policy is loaded

Validate:

1. If First Match Code Groups are used and the site does not have documentation regarding their use of First Match Code Groups then this is a finding.
2. Ask the System Administrator to verify the CAS policy is loaded is not the default policy. The CLR will load the default CAS policy when the policy file is corrupted. If the security policy is not loaded this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0032	First Match Code Groups	
Reference:	.NET Framework Security Guide pg 59	IA Control: DCSL-1

3.4.9.32 APPNET0033: File Code Groups, Net Code Groups

Description: The File Code Groups and Net Code Groups are used to establish directory access and web site connections respectively by the application

Applies to: Version 1.0; Version 1.1; Version 2.0

If a site uses any File or Net Code groups besides the four default groups (*Intranet_Same_Directory_Access, Intranet_Same_Site_Access, Internet_Same_Site_Access, Trusted_Same_Site_Access)* ask for documentation of the .NET Framework Security Policy organization.

Check for the CAS configuration file for any non-default File or Net Code groups.
Look for any entries in the configuration file similar to the following:
<CodeGroup class="System.Security.Policy.NetCodeGroup
<CodeGroup class="System.Security.Policy.FileCodeGroup

Configuration File Location:
Enterprise Level: %WINDIR%\Microsoft.Net\Framework\vx.y.zzzz\config\EnterpriseSec.config
Machine Level: %WINDIR%\Microsoft.Net\Framework\vx.y.zzzz\config\Security.config
User Level: %APPDATA%\Microsoft\CLR Security Config\vx.y.zzzz.ww\Security.config
The vx.y.zzzz are version strings for the .Net Framework

v1.0.3705
v1.1.4322
v2.0.50727 (v2.0.50727.832 for the user level)

Validate any use of any non-default File or Net Code Groups was intentional and the implementation of the Code Group is correct. Ask the System Administrator to validate the security policy is loaded

Validate:

1. If File or Net Code Groups are used and the site does not have documentation regarding their use then this is a finding.
2. Ask the System Administrator to verify the CAS policy is loaded is not the default policy. The CLR will load the default CAS policy when the policy file is corrupted. If the security policy is not loaded this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0033	File Code Groups, Net Code groups	
Reference:	.NET Framework Security Guide pg 61	IA Control: DCSL-1

3.4.9.33 APPNET0035: Level Final Code Group Attribute

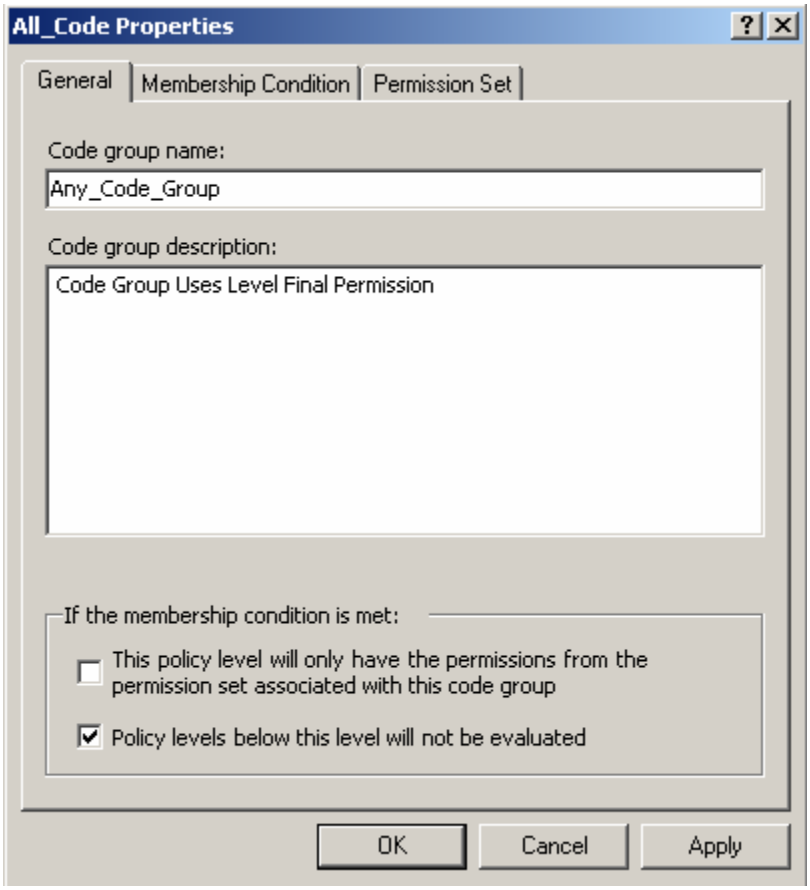
Description: The Level Final Code Group Attribute prevents permission sets farther down in the Code Group hierarchy from being applied to the assembly.

Applies to: Version 1.0; Version 1.1; Version 2.0

MSCORCFG.MSC: Review the .Net configuration for any non-default code groups that use the Level Final permission.
The Level Final permission is used when selecting the “Policy levels below this level will not be evaluated” box.

Validate:

1. If Level Final Code Groups are used and the site does not have documentation regarding their use then this is a finding.



Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0035	Level Final Code Group Attribute	
Reference:	.NET Framework Security Guide pg 70	IA Control: DCSL-1

3.4.9.34 APPNET0041: Zone Membership Condition

Description: The Zone Membership Condition determines policy level based on the URL zone of the application origin.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Review the caspol.exe listing for all code groups. Search for all instances of the text "zone". Note the code group name preceding the attribute.

MSCORCFG.MSC: For each policy level (*Enterprise*, *Machine*, *User)*, expand all Code Groups, expand all child code groups, right-click on each non-default code group in the left-hand frame. (Default code groups are *All Code* at all policy levels and *My_Computer_Zone*, *LocalIntranet_Zone* including *Intranet_Same_Site_Access* and *Intranet_Same_Directory_Access*, *Internet_Zone* including *Internet_Same_Site_Access*, *Restricted_Zone*, and *Trusted_Zone* including *Trusted_Same_Site_Access* under the Machine

policy level.) Select Properties and then select the Membership Condition tab. View the selection listed in the *Choose the condition type for this code group*.

Validate:

1. If a *Zone* membership condition is used for a non-default code group this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0041	Zone Membership Condition	
Reference:	.NET Framework Security Guide pg 96	IA Control: DCSL-1

3.4.9.35 APPNET0045: Administering CAS Policy

Description: The use of the CAS policy can be enabled or disabled on the system.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Review the caspol.exe listing for all permission sets. Search for the occurrence of Security *is Off*.

Validate:

1. If CAS Policy has been disabled then this is a finding.

Category:	CAT I	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0045	Administering CAS Policy	
Reference:	.NET Framework Security Guide pg 121	IA Control: DCSL-1

3.4.9.36 APPNET0046: Administering the Windows Environment for Test Root Certificates

Description: The Windows system may be configured to allow use of certificates that are designated as being for test use.

Applies to: Version 1.0; Version 1.1; Version 2.0

SETREG.EXE: Review the setreg.exe output for the text *Trust the Test Root.*

```
1) Trust the Test Root.......................... FALSE
```

Validate:

1. If *Trust the Test Root* is set to TRUE on a production system then this is a finding.
2. If *Trust the Test Root* is set to TRUE on a development system and the IAO has not approved the setting then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0046	Administering the Windows Environment for Test Root Certificates	
Reference:	.NET Framework Security Guide pg 127	IA Control: DCSL-1

3.4.9.37 APPNET0047: Administering the Windows Environment for Expired Certificates

Description: The Windows system may be configured to check the application for use of expired certificates.

Applies to: Version 1.0; Version 1.1; Version 2.0

SETREG.EXE: Review the setreg.exe output for the text *Use expiration date on certificates*.

```
2) Use expiration date on certificates.......... TRUE
```

Validate:

1. If *Use expiration date on certificates* is set to FALSE on a production system then this is a finding.
2. If *Use expiration date on certificates* is set to FALSE on a development system and the IAO has not approved the setting then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0047	Administering the Windows Environment for Expired Certificates	
Reference:	.NET Framework Security Guide pg 127	IA Control: DCSL-1

3.4.9.38 APPNET0048: Publisher Membership Condition

Description: The Publisher Member Condition requires member code to be certified using certificates originating from a trusted source.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Review the caspol.exe listing for all code groups. Search for all instances of the text *Publisher*. Note the code group name preceding the attribute.

MSCORCFG.MSC: For each policy level (*Enterprise, Machine, User)*, expand all Code Groups, expand all child code groups, right-click on each code group in the left-hand frame. (Default code groups are *All Code* at all policy levels and *My_Computer_Zone, LocalIntranet_Zone* including *Intranet_Same_Site_Access* and *Intranet_Same_Directory_Access, Internet_Zone* including *Internet_Same_Site_Access, Restricted_Zone,* and *Trusted_Zone* including *Trusted_Same_Site_Access* under the Machine policy level.) Select Properties and then select the Membership Condition tab. View the selection listed in the *Choose the condition type for this code group*.

Validate:

1. If the *Publisher Membership Condition* is used on a Non-default Code Group and the use of that Publishers certificate is not documented and approved by the IAO then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0048	Publisher Membership Condition	
Reference:	.NET Framework Security Guide pg 128	IA Control: DCSL-1

3.4.9.39 APPNET0049: Administering the Windows Environment for Revoked Certificates

Description: This checks the setting that determines whether certificates are checked for revocation status.

Applies to: Version 1.0; Version 1.1; Version 2.0

SETREG.EXE: Review the setreg.exe output for the text *Check the revocation list*

.

```
3) Check the revocation list..................... TRUE
```

Validate:

1. If *Check the revocation list* is set to FALSE on a production system then this is a finding.
2. If *Check the revocation list* is set to FALSE on a development system and the IAO has not approved the setting then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0049	Administering the Windows Environment for Revoked Certificates	
Reference:	.NET Framework Security Guide pg 128	IA Control: DCSL-1

3.4.9.40 APPNET0050: Administering the Windows Environment for Unknown Certificate Status

Description: The settings reviewed in this check determine the handling of certificates with differing unknown statuses due to temporary unavailability of a certificate verification service. For example, certificate verification that is dependent on real-time access to a certificate status server could be unavailable due to a break in network communications.

Applies to: Version 1.0; Version 1.1; Version 2.0

SETREG.EXE: Review the setreg.exe output for the text *Offline revocation server OK (Individual), Offline revocation server OK (Commercial), Java offline revocation server OK (Individual), Java offline revocation server OK (Commercial).*

```
4) Offline revocation server OK (Individual)..... FALSE
5) Offline revocation server OK (Commercial)..... FALSE
6) Java offline revocation server OK (Individual) FALSE
7) Java offline revocation server OK (Commercial) FALSE
```

Validate:

1. If any of the settings listed above are set to TRUE on a production system then this is a finding.
2. If any of the settings listed above are set to TRUE on a development system and the IAO has not approved the setting then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0050	Administering the Windows Environment for Unknown Certificate Status	
Reference:	.NET Framework Security Guide pg 129	IA Control: DCSL-1

3.4.9.41 APPNET0051: Administering the Windows Environment for Time Stamped Certificate Revocation

Description: This Windows setting determines whether the system requires certificates to be time stamped to verify the certificate is current.

Applies to: Version 1.0; Version 1.1; Version 2.0

SETREG.EXE: Review the setreg.exe output for the text *Invalidate version 1 signed objects* and *Check the revocation list on Time Stamp Signer*. If the version of Windows is earlier than Windows 2003, then do not include the value check for *Invalidate version 1 signed objects*.

```
8) Invalidate version 1 signed objects.......... TRUE
9) Check the revocation list on Time Stamp Signer TRUE
```

Validate:

1. If any of the settings listed above are set to FALSE on a production system then this is a finding.
2. If any of the settings listed above are set to FALSE on a development system and the IAO has not approved the setting then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0051	Administering the Windows Environment for Time Stamped Certificate	
Reference:	.NET Framework Security Guide pg 130	IA Control: DCSL-1

3.4.9.42 APPNET0052: Strong Name Membership Condition

Description: The Strong Name Membership condition requires that member assemblies be defined with Strong Names.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Review the caspol.exe listing for all code groups. Search for all instances of the text *StrongName*. Note the code group name preceding the attribute.

MSCORCFG.MSC: For each policy level (*Enterprise, Machine, User),* expand all Code Groups, expand all child code groups, right-click on each non-default code group in the left-hand frame. (Default code groups are *All Code* at all policy levels and *My_Computer_Zone, LocalIntranet_Zone* including *Intranet_Same_Site_Access* and *Intranet_Same_Directory_Access, Internet_Zone* including *Internet_Same_Site_Access, Restricted_Zone,* and *Trusted_Zone* including *Trusted_Same_Site_Access* under the Machine policy level.) Select Properties and then select the Membership Condition tab. View the selection listed in the *Choose the condition type for this code group*.

Validate:

1. If a Strong Name membership condition is assigned to a Code Group ensure the private key is adequately protected by the software developer. Ask the System Administrator how the private keys are protected. If the private key is not adequately protected then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0052	Strong Name Membership Condition	
Reference:	.NET Framework Security Guide pg 132	IA Control: DCSL-1

3.4.9.43 APPNET0054: Administering CAS Policy for Group Names

Description: The use of duplicate code group names within a CAS policy can lead to mis-assignment of permissions.

Applies to: Version 1.0; Version 1.1; Version 2.0

CASPOL.EXE: Review the caspol.exe listing for all code groups. Review all code group names. The code group names follow a sequential number at the far left of the file or screen. Code attributes for the code group are numbered and indented below each code group name.

MSCORCFG.MSC: For each policy level (*Enterprise, Machine, User),* expand all Code Groups, expand all child code groups, and review all code group names.

Validate:

1. If non-unique Code Group names are used then this is a finding.

Category:	CAT III	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0054	Administering CAS Policy for Group Names	
Reference:	.NET Framework Security Guide pg 145	IA Control: DCBP-1

3.4.9.44 APPNET0055: Administering CAS Policy and Policy Configuration File Backups

Description: CAS Policy and CAS Policy Configuration files are required for a complete system baseline and disaster recovery event.

Applies to: Version 1.0; Version 1.1; Version 2.0

Validate:

1. Ask the System Administrator if all CAS policy and policy configuration files are included in the system backup. If they are not then this is a finding.
2. Ask the System Administrator if the policy and configuration files are backed up prior to migration, deployment, and reconfiguration. If they are not then this is a finding.

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0055	Administering CAS Policy and Policy Configuration File Backups	
Reference:	.NET Framework Security Guide pg 164	IA Control: CODB-1, CODB-2

3.4.9.45 APPNET0060: Remoting Services Authentication and Encryption

Description: The *typefilterlevel="Full"* attribute allows unfiltered code to access system resources.

Applies to: Version 1.0; Version 1.1; Version 2

Windows Explorer: From Windows Explorer, browse to the Microsoft.NET\Framework\<version>\CONFIG directory. Use the Search function to locate the text: *typefilterlevel="Full"* in all *.config files.

Interview: Ask the System Administrator what encryption and authentication methods are in place for the remoting channels.

Validate:

1. In Version 1.0 of the .NET Framework if authentication and encryption are not used for all remoting channels then this is a finding.
2. In Version 1.1 or 2.0 of the .NET Framework if authentication and encryption are not used for all remoting channels when the *typefilterlevel="Full", then this is a finding.*

Category:	CAT II	Level: Gold MCL: 1-CSP;2-CSP;3-CSP
SDID : APPNET0060	Remoting Services Authentication and Encryption	
Reference:	.NET Framework Security Guide pg 202	IA Control: DCSL-1

4. VMS 6.0 Process and Procedures

4.1 System Administrator

The following instructions are to be used by system administrators for updating the status of the .NET Framework vulnerabilities in VMS 6.0.

1. Log in to VMS 6.0.
2. Select *Asset/Finding Maint.* from the menu in the left panel.
3. Select *Assets/Findings* from the sub menu.
4. In the right panel navigate through the tree to the appropriate location and expand the *Computing* branch of the tree.
5. If the asset you wish to update does not exist the following steps will allow you to create a new asset. If it does exist, proceed to Step 7.
6. Press the *Create Computing Asset* button located next to the *Computing* tree branch.
7. Verify that the identifying information that appears to the right is correct for the asset you wish to update. If you are creating a new asset you should fill this information in now. Please note that any information entered or changed is not saved until you press the *Save* button.
8. If creating a new asset, update the *Asset Posture* to include the correct Operating System and applications as appropriate.
9. Add the appropriate .NET Framework elements to the *Asset Posture*.
    i. Select the *Asset Posture* tab.
    ii. Expand the *Computing* branch.
    iii. Expand the *Application* branch.
    iv. Expand the *MSdotNetFramework* branch.
    v. Select the appropriate Framework version(s).
    vi. Press the >> button to add the selected elements to the asset posture.
    vii. Press the *Save* button.
10. The selected .NET Frameworks will now be targets under the application and you may now update the status of the vulnerabilities.

4.2 Reviewer

The following instructions are to be used by a reviewer for updating the status of the .NET Framework vulnerabilities in VMS 6.0.

1. Log in to VMS 6.0.
2. Select *Asset/Finding Maint.* from the menu in the left panel.
3. Select *Assets/Findings* from the sub menu.
4. In the right panel navigate through the tree to the appropriate *Visit* and expand the *Computing* branch of the tree.
5. If the asset you wish to update does not exist the following steps will allow you to create a new asset. If it does exist, proceed to Step 7.
6. Press the *Create Computing Asset* button located next to the *Computing* tree branch.
7. Verify that the identifying information that appears to the right is correct for the asset you wish to update. If you are creating a new asset you should fill this information in now. Please note that any information entered or changed is not saved until you press the *Save* button.

8. If creating a new asset, update the *Asset Posture* to include the correct Operating System and applications as appropriate.
9. Add the appropriate .NET Framework elements to the *Asset Posture*.
    i. Select the *Asset Posture* tab.
    ii. Expand the *Computing* branch.
    iii. Expand the *Application* branch.
    iv. Expand the *MSdotNetFramework* branch.
    v. Select the appropriate Framework version(s).
    vi. Press the >> button to add the selected elements to the asset posture.
    vii. Press the *Save* button.
10. The selected .NET Frameworks will now be targets under the application and you may now update the status of the vulnerabilities.